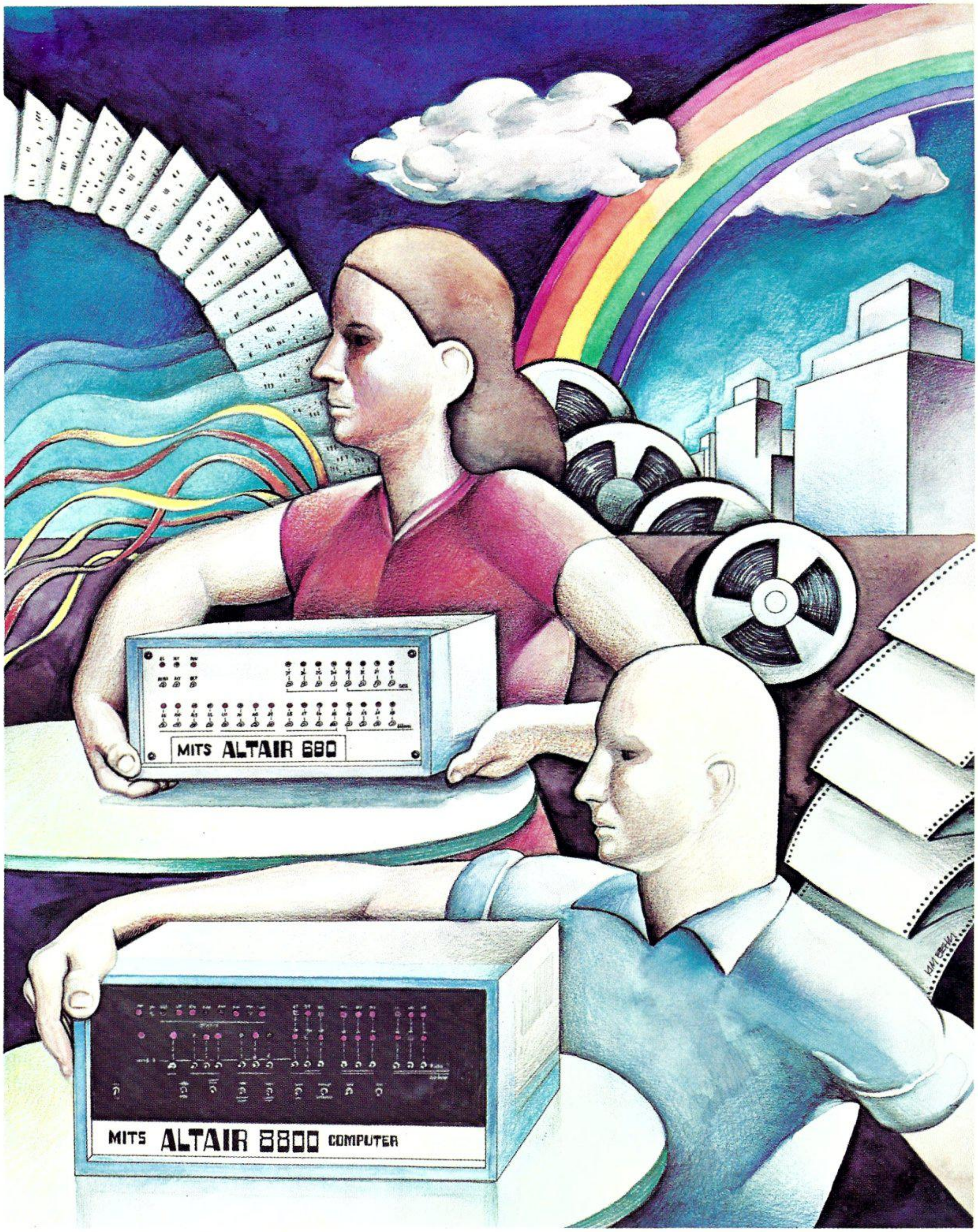
VOL. 1 ISSUE 1 DECEMBER 1975

SCCS INTERFACE

THE OFFICIAL PUBLICATION OF THE SOUTHERN CALIFORNIA COMPUTER SOCIETY


MITS ALTAIR 680
MITS ALTAIR 8800 COMPUTER
KIM BEHM

THE PEOPLE COMPUTERS

Painting by Kim Behm

FEATURES

Toward the Design of a Micro-operating System *Lloyd Rice*
Teleprinter Maintenance / Repair / Preventative Maintenance
Altair Switch Memory *S. Wilcox*
1 + 1 = 10? *Jeff Landry*
Soup Up Your T.V. Typewriter *Mike Thompson*

Price $1.25

OFFICERS

Ward Spaniol	President
Lou Fields	Vice President
Don Tarbell	Secretary
Hal Lashlee	Treasurer

BOARD OF DIRECTORS

Pat Amornvitikivacha	'75-'76
Bob Baskin	'75-'76
Art Childs	'75-'76
Ray Delong	'75-'76
Lou Fields	'75-'76
John French	'75-'76
Roy Fulton	'75-'76
Lois Heiser	'75-'76
Chris King	'75-'76
Rubin Loshak	'75-'76
Marvin Perlman	'75-'76
Larry Press	'75-'76
Ron Rogers	'75-'76
George Rowe	'75-'76
Jerry Silver	'75-'76
Mike Stern	'75-'76
Don Tarbell	'75-'76
George Tate	'75-'76
Carl Todd	'75-'76
Pearce Young	'75-'76

SCCS INTERFACE
Editorial Staff

Art Childs, Editor
Scott Wilcox, Hardware Editor
Jeff Landry, Software Editor
Nancy Delong, Associate Editor
Mike Stern, Games Editor
Mike Teener, Bits 'n' Bytes Editor
Larry Press, Applications Exchange Editor

THE SOUTHERN CALIFORNIA COMPUTER SOCIETY

OBJECTIVES

A non-profit organization whose purpose is:

A. To exchange and disseminate information among the Society members concerning the computer arts and sciences.

B. To provide a technical assistance to other members of the Society in those computer projects which are not undertaken for pecuniary gain or profit including but not limited to hardware, software, and computer programming.

C. To publish books, newsletters, magazines, and other periodicals for the benefit and education of the Society members and the general public.

D. To conduct and sponsor seminars, lectures, and courses relating to the computer arts and sciences.

F. To develop and maintain computer centers and laboratory workshops for the members of the Society and the general public including provisions for time-sharing operations.

PUBLICATIONS

SCC INTERFACE, the official publication of the Southern California Computer Society is published monthly. Its content is composed primarily of articles contributed by members of the Society and intended for the reader with an interest in computers for professional or avocational reasons.

MEMBERSHIP

Membership in the Society is open to anyone, regardless of educational background or geographic location. All members are entitled to the rights of meetings, elections, receipt of official publications in a timely manner, and all other benefits as provided by the officers, Board of Directors, and various committees of the Society pursuant to the by-laws of the Southern California Computer Society.

CLASSIFIED ADVERTISING

Each month SCCS Interface will devote some space free, non-commercial advertising by members. This is done as a service to members of the Southern California Computer Society and to help promote communications between computer experimenters and hobbyists throughout the world.

Many SCCS members, being pioneers in the small computer field, often have for sale, on an occasional basis, used computing equipment, or equipment they have designed and built themselves. Those members, as well as those who simply wish to establish contact with individuals with similar or complimentary interests, are encouraged to utilize the member ad privilege subject to the following conditions:

A. Ads must be received by the ad editor of SCCS Interface by the first of the month prior to issue month.
B. Ads may list the member's name, home address and phone. No commercial firm names, please!
C. Ads must be typewritten, double spaced and no more than 250 characters (including spaces and punctuation) in length. Ads 250-500 characters will be printed on space-available basis.
D. Ads may be run up to three consecutive issues, however, a specific request must be received for each instance.
E. No free member ads will be accepted for the sale of equipment or services by a commercial profit-oriented individual or organization.

Address all member ads to:
Interface Ad Editor, Nancy Delong, 8005 Denrock Ave., Los Angeles, Calif. 90045

Those non-members wishing to advertise are invited to join SCCS and enjoy the member ad privilege as well as the delivery of SCCS INTERFACE to your mail box.

COMMERCIAL CLASSIFIED

Commercial ads limited to business card format will be accepted at the rate of $25.00 per ad. Payment must accompany order. Submit orders to:
Advertising Department, McPheters, Wolfe & Jones, 6515 Sunset Blvd., Suite 202 Hollywood, California 90028

NOTE: Commercial display advertisers are invited to contact Nancy Jones at the address listed above. For rate information, write, or call her at (213) 469-7789.

PUBLISHER

McPHETERS, WOLFE & JONES, 6515 Sunset Blvd., Suite 202, Hollywood, California 90028

All editorial content and effort are contributed at no cost to the Society. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the Southern California Computer Society. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes an endorsement of the product or products by this magazine or by the Southern California Computer Society.

ADVERTISING

Direct all advertising inquiries to:

Advertising Department
SCCS INTERFACE
c/o McPHETERS, WOLFE & JONES
6515 Sunset Blvd., Suite 202
Hollywood, California 90028

CONTRIBUTIONS

They must be accompanied by return postage and we assume no responsibility for loss or damage thereto. Any material accepted is subject to such revision as is necessary in our sole discretion to meet the requirements of this publication. The act of mailing a manuscript and/or material shall constitute an express warranty by the contributor that the material is original and in no way an infringement upon the rights of others.

SCCS INTERFACE

The Official Publication of the Southern California Computer Society

Vol. 1 Issue 1 December 1975

FEATURES

TOWARD THE DESIGN OF A MICRO-OPERATING SYSTEM .... Pg. 3
By Lloyd Rice

TELEPRINTER MAINTENANCE/REPAIR/ PREVENTATIVE MAINTENANCE .... Pg. 3
By Cliff Sparks

SOUP UP YOUR T.V. TYPEWRITER................................ Pg. 8
By Mike Thompson

1 + 1 = 10?.. Pg. 11
By Jeff Lendry

FEATURETTES

ALTAIR ALTERATION................. Pg. 7
By Roger Hall

ALTAIR SWITCH MEMORY .... Pg. 14
By S. Wilcox

HARDWARE NOTES .... Pg. 17
By S. Wilcox

DEPARTMENTS

UPDATE.................Pg. 2

BITS & BYTES ....Pg. 6

APPLICATIONS EXCHANGE ....Pg. 14

GAMES & THINGS ....Pg. 18

INTERFACIAL ....Pg. 26

LETTER TO THE EDITOR . Pg. 27

Happy Holiday Season

UPdate

What has happened and is happening with SCCS

From the Facility Committee — plans for some great things are in progress. A FOUR PART TIME-SHARING SYSTEM, A LIBRARY AND DATA BASE REPOSITORY, AN ELECTRONICS LAB AND WORK SHOP, AND A GENERAL OFFICE AND MEETING PLACE. Larry Press, a member of that committee, invites those interested in helping or those with ideas and suggestions, to call him at (213) 399-2083.

The Board of Directors has been meeting weekly since its inception. The long hours are beginning to produce results, most of which are concerned with the various committees charged with the responsibilities of guiding the club's efforts in several areas: obtaining resources and directing their use, establishing and staffing a hardware center, obtaining the equipment to set up a time-sharing facility, writing and publishing the club magazine, and planning the monthly meetings. Other projects are in the idea stage. Members are invited to participate in these projects and to serve on the various committees. Anyone interested may call Ruben Loshak at (213) 338-8589 to be put in contact with the group of choice.

BRAND X.......

Update is the section of SCCS INTERFACE reserved for club news, announcements, and similar items. In this section we'll be printing SCCS club news *and* the club news (including organizing efforts) of any other non-profit computer club.

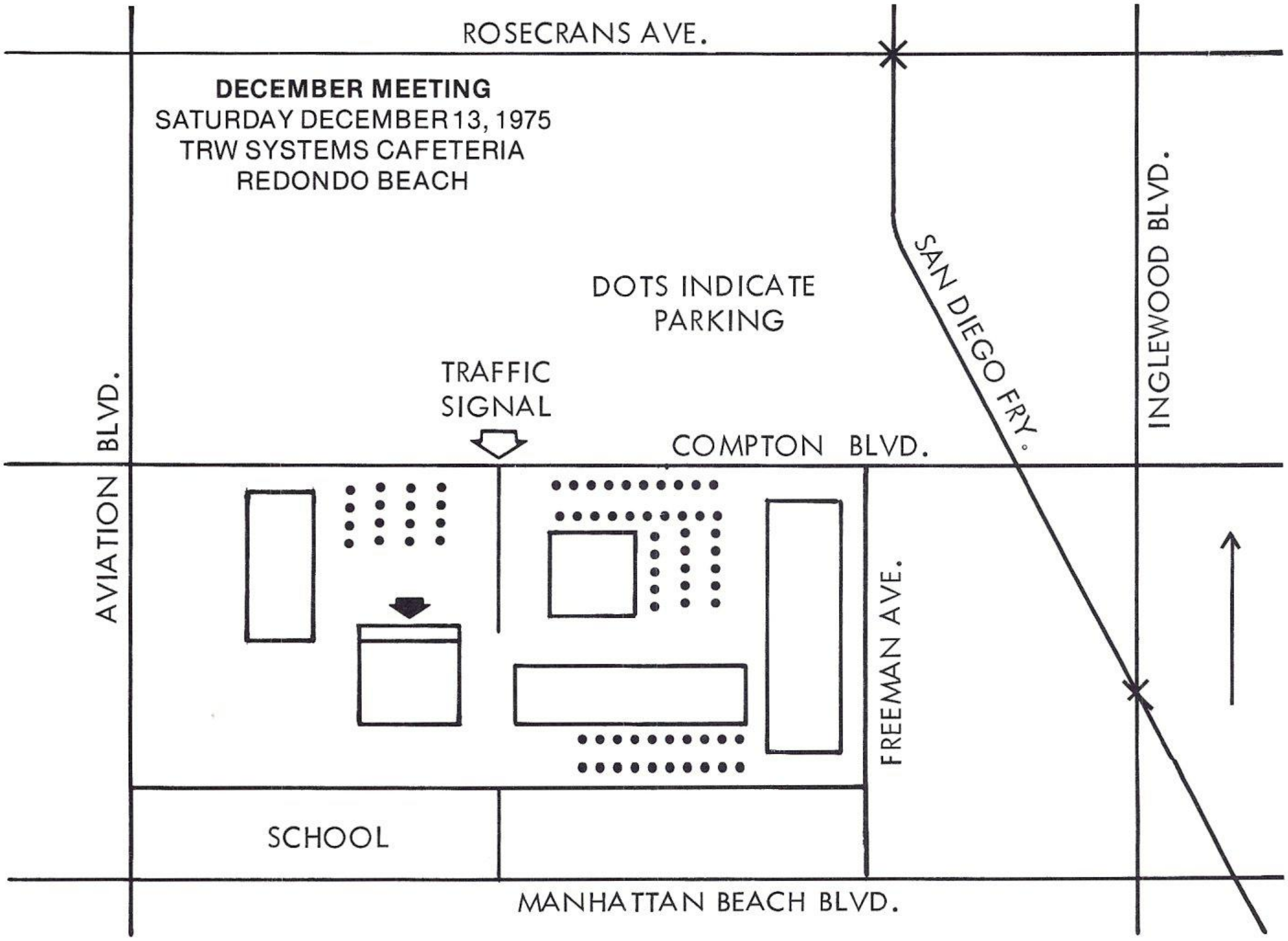
To participate, simply send your data to Art Childs, 335 North Adams St., Glendale, California 91206.

To make a specific deadline, copy should be received by the first day of the month prior to the issue month.

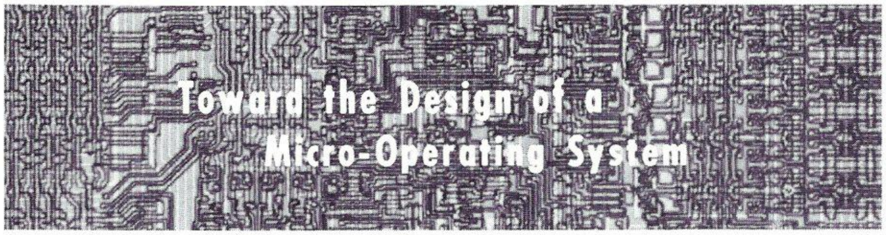
Please keep announcements brief to help conserve space. Clubs are invited to submit a description and history which will be printed when possible.

Persons attempting to organize a computer club in a particular area are also invited to submit details — we'll help you all we can.

Editor



DECEMBER MEETING
SATURDAY DECEMBER 13, 1975
TRW SYSTEMS CAFETERIA
REDONDO BEACH



Toward the Design of a Micro-Operating System

Lloyd Rice
821 Pacific St. #4
Santa Monica, CA 90405

To design nearly any chunk of software is an exercise in juggling conflicting, competing demands, and designing an operating system is probably the example par excellence of this situation. The most severe tradeoffs are memory space and simplicity versus the many nifty features you would really like to have. I will begin this discussion with an outline of operating system functions which could be considered in designing a microprocessor operating system. I intend this outline to be a starting point, open for discussion, around which a real system will grow and develop. Following the outline, I will expand the discussion of each function and indicate how that function would be applicable to a range of different configurations.

At this stage of planning, I intend to keep the remarks general rather than specific to any particular brand of processor. On the other hand, It will become clear that decisions regarding the operating system depend heavily on the number and types of peripheral devices available. I would like to explore the question of whether a "standard" operating system could be laid out which would be suitable for a variety of configurations. Please keep notes on any comments and send them to me.

Operating System Functions

A. General Monitor Functions
  1. Keyboard interaction for control commands
  2. Inform user of system level error conditions
  3. Debugging aids
  4. Create a "run file" by saving current memory contents
  5. Load and start a "run file" (previously compiled, assembled, saved, etc.)

B. Other System Functions
  1. System I/O (character, word, binary block)
  2. Device handler maintenance
  3. Misc. format conversions, frequently needed routines, etc.
  4. File directory operations
     a. file look-up
     b. create a new file
     c. close an open file
     d. delete
  5. Passing file parameters to "run" jobs

(Continued)

Teleprinter Maintenance

Repair

Preventative Maintenance

By Cliff Sparks

This series of articles will provide the reader with a working knowledge of teleprinter repair as applied to the model 33 ASR/KSR.

In the early 1920's, a clever Frenchman by the name of Baudot introduced the first teleprinter, as we would recognize it today. Prior to that time coded tape and special purpose devices (ticker tape) appeared as early as 1847 on the New York Stock Exchange. The early tickers received data at 100 characters per minute.

Communications centers were established in Denver, Los Angeles, New York City and Atlanta as early as 1915. A Kleinschmitt 2B printer was used in message centers to emboss a typed message on a quarter inch tape. The tape was then torn into selected lengths and glued onto a Western Union form, "the telegram".

Cable transmission lines between New York City and Denver proved unreliable and more often, non-existent. Equipment was designed, redesigned and over engineered into oblivion. Subsequently the best, longest, and most economical transmission media was initiated . . . the railroad track. The track replaced wire and the teleprinter equipment was tapped to it, thereby creating the series loop circuit.

A simple current loop between teleprinter battery and ground insured the operator that the railroad track was intact and circuit continuity existed. Removed or damaged track was quickly discovered if the current loop was disturbed.

Perhaps the reader who has muttered curses over a printer with parallel inputs to a series output can now understand why the standard was adopted. Today's teleprinter machines are designed for two wire point-to-point communications via a simple series current loop.

Between 1920 and 1936 the use of the teleprinter machine as a prime communications device exploded. A similar situation existed then as it does today with the ever-popular minicomputer systems. Vast networks were developed using the new teleprinter equipment and such firms as ITT, AT&T, Western Union and the U.S. Army Signal Corps found new and challenging frontiers by developing new networks.

The next article will introduce the reader to each major section of the 33ASR/KSR and explain their purpose.

A very early model (Kleinschmitt 2C) strip teleprinter device will be on display at the next SCCS meeting.

(Continued from Page 3)

C. Multi-task Environment
  1. Job demand handling
    a. foreground-background allocation
    b. time-out of running job
    c. interrupt by higher priority job
  2. Job swapping and job status maintenance

Where does the operating system live?

Memory costs have usually been fairly high relative to the total system cost and this has tended to limit the address space as designed into the CPU. A PDP-8, for example, could basically address 4K, with a 3-bit extension register allowing expansion to an absolute maximum of 32K. In later 16-bit minis and most microprocessor chips, it is not unusual to have an address space of 65K. One result of this is that it becomes feasible to dedicate 1 or 2 K or even more of that address space to the operating system, eliminating the need to swap it in from a peripheral storage device. Where address space is more limited and more valuable, a readable, "system device" is necessary.

The operating system is not all read-only code. Tables must be kept up-to-date and available which contain device status, job status, I/O buffers, possibly file directory information and other various tidbits. If a reasonably fast, random access read/write device is available, many of these tables can be kept there but in any case, some addressable memory must be reserved for such information. This portion of the system is often called the resident. In many system layouts the highest page or two of memory is reserved for these system information tables. The bulk of the operating system, however, is read-only and can reside in a ROM, whether that is directly adressable or accessed as a peripheral.

General Monitor Functions

I view the monitor as that portion of the operating system which interacts directly with the user. This includes keyboard commands to control the entire software system and some sort of indication of software error conditions. The essential function of the monitor is keyboard control of the system. Thus, this portion of the monitor includes a keyboard input routine, a command line buffer, possibly some degree of command line editing, and a scanner to match up the command input with a table of valid commands. Most commands to the system would be executed on a line-by-line basis, with the command text stored in a buffer until the final key is hit, usually a carriage return, which means "do it!". Some commands, on the other hand, will be single keys such as rubout for command editing. The keyboard input routine must watch for these characters for immediate execution. Once a basic command line capability is provided, the range of commands which are actually implemented is fairly flexible. Different operating systems could be classified by the ease with which new commands can be added to the tables and implemented. The simplest systems have a fixed list coded in and are not easily changeable.

I have listed in the outline what I consider to be the most basic, necessary monitor commands for any system. This list could easily be expanded at the user's whim and many possibilities come to mind when you sit down at the console.

In some cases, however, an alternative route could lead to the same ends in a simpler way. If a reasonably fast, random access device is available then special programs can be written to implement most of the potential additions to the monitor list. For example, you might want to delete a particular file on a specified device. The code to read that directory, perform the deletion and rewrite it could be included within the monitor itself, executed by a monitor level DELETE command. Or it could be in a small program which is run to do that specific job. A DELETE program would accept a file specification from the operating system as it was loaded and would proceed to delete that file.

This approach may simplify the command structure, but there is a catch. If another program running under the system needs to delete a file, you either have to duplicate that order or provide a way for programs to call other programs as subroutines — which would be neat, but that could really get hairy. I'll get back to this problem later under other system tasks.

Debugging

In a small, hands-on system one of the important basic monitor functions is some sort of debugging aid. A suitable debugging aid can be used to write simple programs without the use of an assembler. Such an aid could consist of no more than the ability to store numerical values into sequential memory locations via keyboard entry, or it could include editing and job start-up capabilities all the way up to the level of a combined editor — symbolic assembler in-core. Probably the most important features would be the following:

1. Type out or display the contents of any specified memory location
2. Type out or display the contents of the next sequential location
3. Store a specified numerical value in any specified memory location
4. Store a specified numerical value in the next sequential location
5. Begin execution at any specified location
6. Stop execution and return to the monitor keyboard input routine

(Continued on Page 9)

You can buy this microcomputer for $39.95, but...

MC 6800

We would be a bit surprised if you could do anything meaningful without additional hardware and software. **Wave Mate's Jupiter II**™ isn't the kind of microcomputer kit you only stare at . . . when you've completed your Jupiter II just plug in your teletype and you're ready to go. That's because it goes beyond the sum of its high quality parts. It's the ultimate micro kit experience. **In performance, in documentation, in reliability.** First, consider its superb features. It has small **pluggable wire wrapped cards** easily tailored to suit your modifications. **Every IC is socketed** and **100% burn-in tested.** In fact every part including the powerful MC 6800 CPU and the 8K dynamic RAM is guaranteed for 120 days. It has the best software around, **System Monitor** and **Debug programs** (ROM). Includes powerful **text editor** and **Motorola compatible assembler.** And BASIC at no extra cost. Because we've been making microcomputer systems for over 4 years, we can offer you the broadest line of interfaces including TV terminal and dual audio cassette. Impressive. And yet the grandest feature is the experience of completing a kit that works. Guaranteed.

Wave Mate

ACT NOW AND SAVE.
Introductory price $1225. (Assembled $1885.)
(Good until February 1, 1976)

☐ Send Jupiter II Microcomputer Kit. Includes:

- Modular plug-in power supply
- 9 module PC backplane
- CPU module
- System monitor module
- 8K dynamic RAM module
- Serial RS 232 communication interface module
- Front panel module
- Front panel
- Wire; cut, stripped, color coded
- Rack mount module cage
- Wire wrap tool
- Wire unwrap tool
- Cables, connectors, all other necessary hardware
- Software (editor, debug, assembler, BASIC)
- Assembly manuals
- Operators manuals
- Theory of operation manuals
- Annual membership in users group

☐ Send details on kit-a-month club
☐ Kit ☐ Assembled
☐ RS 232C conversion kit (teletype) $47.85
☐ I'm not convinced; send me 101 reasons why.
☐ Please send free Jupiter II System catalogue.

☐ Check enclosed for $ __________
Include $10.00 for postage and handling. California residents add 6% tax Delivery 60 days ARO
☐ Master Charge # __________
☐ Bank Americard # __________
Credit card expires on __________

Name __________
Address __________
City/State/Zip __________

WAVE MATE
1015 West 190th Street Gardena, California 90248
Telephone (213) 329-8941

Wave Mate

WARRANTY
120 days on all parts, materials and workmanship.

Bits and Bytes

by Mike Teener

It's always going to happen. You've just bought your Altair 8800 with 1K of memory and a serial I/O card to hook to your reconditioned teletype . . . and then MITS brings out the 6800 with almost as much power for half the price you paid. And you always wanted the 6800 more than the 8080 anyway.

Or maybe you're a real homebrewer going the cheapest route possible with an 8008 chip for $30. You've got your system almost complete and then you open up your new BYTE and find you can get an 8080 or 6800 for $40 or a 6501 for $20.

It's a hard life, being a computer hobbyist, and something we're always going to be plagued/blessed with is rapid technological advance and the resulting fast obsolescence. I've got a digital clock I built five years ago that cost me $50 for the fifty or so TTL MSI and SSI chips and another $20 for the display. I keep it around to remind me that no matter how up-to-date my projects are at the time I start them, they're going to be antiques pretty soon — maybe before they're finished.

So what can we do? Nothing, I'm afraid. Part of my real job is to keep people aware of new developments in the computer industry, and I'm pretty good at that, even if I have to say so myself. Still, if I don't get floored by some new product announcement at least once a week, things are pretty slow.

So live with it. Glory in it. Enjoy being part of such a fast growing field. Keep playing with your 8008 system . . . it's a tougher project anyway . . . sharpens the mind. So you paid twice as much for your 2102's as they're going for now, you got them that much sooner. Besides, all those people who are buying 6800's right now are going to kick themselves in six months or less when the hundred-buck computers start coming down the line. Me? I'm going to get a 6800 anyway.

****

BASIC is OK, but what I really want is APL. Then I can have a contest every month on "what does this program do?" For those of you who don't know, APL is an incredibly compact and powerful interactive language that uses the English alphabet, the Greek alphabet, and most of the special symbols in the known universe. I can write one line programs that find the greatest common divisor of an integer or sorts a string of characters in alphabetic (or reverse, or random) order. Get a version of that up on your 8080 or 6800 and I'll take you out to dinner.

****

This section of INTERFACE is for all those little pieces of information that you might have that someone else might like to know. So write already! If you have a problem that others might be able to solve, write! If you have a problem and nobody can solve it, write! We aim to please . . .

****

One more thing, there is a new micro from Texas Instruments, the 9900. This is a full 16-bit NMOS processor in a dip with a mind-boggling **64** pins. It's supposed to be very fast and powerful, but I won't have complete details until next month. It's going to be pretty cheap, too. A packaged version with 8K bytes of RAM, clock, 2 I/O busses (slow and DMA) $800 in unit quantities. A little much for the average hobbiest, but that CPU chip is going to be available soon for lots less.

Interesting items from the trade press:

NATIONAL SEMICONDUCTOR (PACE, IMP) has let it be known that they are going after the 'mechanical logic' market with their new series of PMOS microprocessors. This means washers and driers, sewing machines, cars, and even coin machines will be microprocessor controlled. They are going to be **cheap.**

MOS TECHNOLOGY has a couple of interesting chips, the 6530 TIM and KIM ROM/RAM/IO/interval timer circuit. One 40 pin unit has 64 bytes of RAM, 2 eight-bit bi-directional programmable I/O ports, programmable interval timer (with interrupt) and 1K bytes of ROM pre-programmed with one of two resident monitors — the KIM (Keyboard Input Monitor) for control of a teletype, a 6 digit, 7 segment display, and a 22-key keyboard — and the TIM (Teletype Input Monitor) which is similar, but without the keyboard/display control and with the addition of paper tape control. This means you only need two chips to get the system up, a 6502 and a 6530. Chuck Peddle and gang strike again.

D E C is bringing out yet another version of the venerable PDP-8, the SUPER-8. It's got about 60 new instructions allowing **direct** addressing to 32K words (oh, where have all the pages gone?), and double precision arithmetic (you can count beyond 4096!) and 72 and 36 bit floating point. All in a package roughly 50% larger than the Altair 8800. Price? Around $5K completely optioned up.

****

LARGE SCALE SYSTEMS

by Ralph Klestadt

What is a large scale system? Many people have an idea of what one is, but can not adequately define it. Can you measure it by the number of lights on the console? No, the recent 370's have but one light on the console, while the Altair has well over 30. Can you measure whether it is a large scale system by the number of peripherals? No, a Nova mini can have as many peripherals as one of the big DEC system-10's. By physical size? or execution speed? No, a PDP-11/70 is larger and faster than some IBM System/360's. By the size of the accumulators? No, the PDP-11/45 may have some 64 bit AC's, while a 360 has 32 bit AC's. Perhaps by the number of users? Certainly not, inventory minicomputers may have several hundred terminals attached to them, while the Pentagon often uses large scale systems on a one user basis. By number of installations? No, for minicomputers far outnumber large scale systems.

No, you cannot measure a large scale system by any one of these traits, but rather by a combination of all of them. Too much emphasis has been put on mini's and micro's in hobbyist magazines; let's take a look at the giants of the DP industry. In future articles, we will be taking a look at different types of systems such as CDC, DEC, and IBM models, different types of processing such as remote job entry, timesharing, batch, and real time. We will be taking a look at some different communications networks concerning the large scale system, and we will explore some of the applications of these systems.

Development of languages will also be discussed, along with any other topics concerning this area of data processing that you wish to send in. We will try to examine as much of this portion of the industry as possible, and hope that you will have some suggestions also.

* * *

ALTAIR ALTERATION
by Roger Hall

EDITORS NOTE: The following 2 articles aren't intended as criticism of the Altair 8800. It seems that perfection is often a typical computer hobbyist characteristic, and what perfectionist would tolerate noise on a DC suppy line?

Further elimination of power line interference can be accomplished by mounting the power switch on the rear panel, just below the line cord. Why there? A number of people have expressed dismay about the front-mounted location due to accidentally hitting the power switch and watching their programs disappear.

Okay, so where to put it? Can't stick it in one of those two neat little holes below the connector cutouts, because the line garbage will still infiltrate all your input and output lines. So, the shortest path is the easiest, and

(Continued on Page 17)

Memory Mania!

4K — $215!
8K — $402!
12K — $562!
16K — $696!

IT'S ALL DYNAMIC MEMORY!

The SPHERE Memory board. Is designed to hold 16K of Memory on one board. 2107A-8 memory chips are used on this board and are designed in 4K increments to allow choice of 4, 8, 12, or 16K. Each memory chip is socketed and is delivered to you completely assembled and tested. The PC memory board connects to the CPU board tri-state buffered bus cables. The memory locations are strap selectable as to order of addressing. Those memory mania purchasers of 12 or 16K who have, or who buy, any SPHERE Computer System will get absolutely free the SPHERE extended BASIC compiler. All documentations will be included.

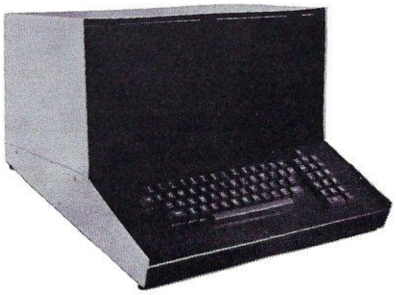
THE MORE THE MEMORY...THE MERRIER.

AND...If you're not into computers yet...Look at this:

$622	CPU Board: 8 bit parallel microprocessor with 4K dynamic memory. 1K EPROM containing program development system or Basic language subset or your specific application software. Real-Time Clock, bus cable structure and 16 lines of digital I/O optional.
$207	CRT Board: generates 64 character ASCII character set which is displayed on 16 lines by 32 characters (5 x 7 matrix). Electronics will display on television set or video monitor.
$160	Keyboard: ASCII type alpha/numeric keyboard with numeric and cursor keypad.
$219	Communications Board: includes FSK Modem (110 to 600 baud) originate/answer with cassette or telephone interface includes serial I/O capabilities (110 to 9600 baud) it is EIA, TTL, TTY, current loop I/O and RS 232 compatible.
$160	PIM: Peripheral Interface module, 64 digital I/O board allowing complete interfacing with printer and Floppy Disks Systems.
$96	Power Supply: produces 5 volts at 5 amps, 12 volts at 3 amps, −5 volts at 400ma, −12 volts at 400ma.

Or...you can buy the combination of the above assembled in an attractive chassis with line printer and/or Floppy Disks starting at $1400 for assembled systems, while kits start at $860!

Check with us on our options too, from Light Pens, to Color and B/W graphics terminal, line printers, Disks etc.



SPHERE
CORPORATION
791 South 500 West Dept. 128 (801) 292-8466
Bountiful, Utah 84010

Soup up your tv typewriter

by Mike Thompson
(T.V. Typewriter II Owners)

I. THE CURSOR

Within a minute and a half of completion of my TV typewriter, I found myself cursing at the cursor. Within another 90 seconds I began to feel a barely controllable urge to hit that blinking square with something hard. What was to be done?

The standard circuitry gave me three options: 1) no cursor, 2) a solid square in place of the character the cursor was to indicate, 3) that @(#*$&%- blinking square superimposed over the character. The first was an improvement in annoyance level but not in ease of typing. The second would leave me guessing what was hidden behind the cursor, and if I paid for a 512 character display, I wanted to see all of them. As for number three, I just try looking at a screen with a little blinking square, and I won't have to tell you what's wrong with it. Obviously, some changes had to be made.

Since the cursor couldn't go where the character was, it had to go above or below it. Reading the theory of operation made it clear that it couldn't go above the character (well, at least not as easily).

Examination of the circuitry and explanatory text showed that the cursor was generated by de-enabling the 2513 character generator (IC22) when the following occurred simultaneously: 1) the cursor and address counters matched, i.e., a 'compare' (IC3d pin 11 high). 2) the display was not on lines 8 or 9 (IC12c pin 10 high), and 3) it was not during the vertical retrace interval (IC4b pin 9 high), 4) it was not during the horizontal retrace interval (IC12c pin 12 high).

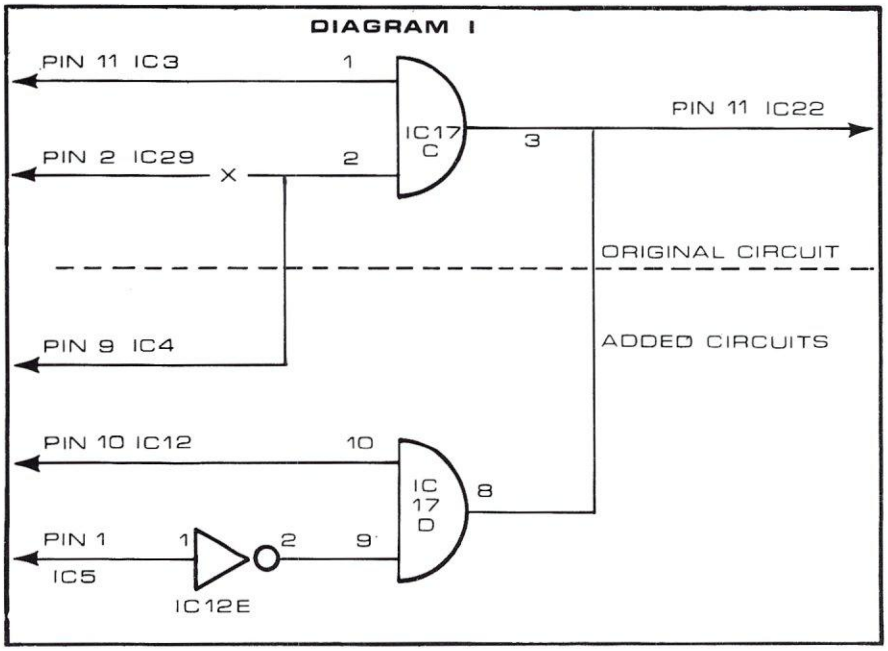
I saw no reason why I couldn't feed the cursor a different, but similar signal, scheduling it to occur on line 8 rather than on 0 through 7. By changing one connection on the printed circuit board and using a couple of extra gates that were lying around on the circuit board, I was able to get a nice little line under the character position where the next character would be entered.

For those of you who share my phobia for blinking squares, here's what you do about it:

1. Cut the foil leading from pin 2 of IC17 to pin 2 of IC29. Cut it on the bottom of the board near IC23.
2. Connect a wire from the through-hole on the foil you just cut near IC16 to the end of R26 nearest the edge of the board. (This connects pin 2 of IC17 to pin 9 of IC4).
3. Connect a wire from pin 3 of IC17 to pin 8 of IC17 (be careful of solder bridges).
4. Connect a wire from pin 9 of IC17 to pin 2 of IC12.
5. Connect a wire from pin 10 of IC17 to pin 10 of IC12.
6. Connect a wire from pin 1 of IC12 to pin 1 of IC5.

Double check that you've caused no solder bridges and make sure that you hook up to the proper IC's. (I goofed a couple of times, but fortunately TTL is tough stuff.)

DIAGRAM I



HOW IT WORKS:

IC17 is an open collector AND gate. When the outputs of IC17c and IC17d are wired together (wire anded), they become 4 input AND gate. When it's 4 inputs are high, the output goes high, causing the enable input of IC22 to go high which disables IC22, the character generator, causing it to output all ones. The four inputs are high when the display is on line 8 (pin 10 if IC14 high and pin 1 of IC5 low — its output inverted by IC12e). There is a compare not during the horizontal retrace (pin 11 of IC3), and it is not during the vertical retrace (IC4 pin 9 is high). This is just the right spot to display a discreet little line.

Hook up everything and apply power. If the picture doesn't come on normally right away, turn it off quickly and look for errors. If all is well, a home will find the cursor, as it may be hard to spot if the screen comes on filled with random characters.

Speaking of random characters, would somebody care to correct the problem of those random characters appearing when power is first applied?

(Continued from Page 4)

Notice that 6. would require some sort of interrupt capability. Most micro chips have a fairly good interrupt capacity which should be used. The alternative is a console system restart whenever it's not going the way you think it should. Additional useful features would be

7. The ability to setup program breakpoints at specific locations
8. The ability to search memory contents for a particular value
9. Search as in 8. but maskable by a specified bit pattern
10. Type out memory contents in various codings, BCD, ASCII, etc.
11. Store values in memory by various codings, BCD, ASCII, etc.

How number 7. is done depends on the particular CPU architecture. It may in fact be satisfied by number 3. above. The essence is that when a preset breakdown is encountered, various register contents such as the accumulator, index registers or status bits are typed out or displayed.

A Binary Loader

Once you get your computer on the air, it won't be long until you're tired of putting programs in by console switches or even by typing in octal or hexadecimal on the keyboard. As soon as you have some peripheral device capable of storing code, you'll want to be able to write out programs and be able to reload them later (shades of the HP-45). If this device is simply the paper tape on the TTY, then the system doesn't have much to do except store the bytes as they come in, detect the end of the tape (maybe manually) and then start the job. Good enough. This is the basic function of a binary loader.

Of course elaborations on the basic plan grow quickly. You may want the starting address punched at the end of the tape so the loader sees it and starts the job automatically. Why not be able to punch a restartable tape directly from the present contents of memory? Now suppose you have more than one peripheral device. You might want to copy one of your paper tapes onto a cassette with a bunch of related programs so you can load any one of them from the cassette later. This implies a common format or format conversion between devices and a directory of some sort on the cassette.

A generalized binary loader capability would include a device independent load-file structure so the device assignment to the loader could specify any peripheral for which standardized I/O was available (see next section on standard I/O). A companion program-save command would write this standard load-file structure onto an assigned device using standardized I/O calls, saving the current contents of specified sections of memory and including a specified job restart address.

General characteristics of system I/O

A very desirable feature of a system is to provide a standardized call for all I/O transfers so that each program is not specifically written for TTY, cassette or whatever, but rather, makes general I/O calls to the system. Specific devices are attached (or "assigned") to the program's I/O needs by the system, usually at the time that program is started, or occasionally these may be reassignable during the program's execution. The cost of not building in this sort of capability is that each program will have to be rewritten if you want it to output to the CRT or to the disk instead of to the TTY, etc.

The problem raised by standardized I/O calls is that since some devices, such as disk, magnetic tape, cassettes, etc. require blocks of data at a time instead of single characters, this blocking (buffering) must be present as a part of the standardized I/O call. This is no problem for compilers, assemblers or most "data processing" tasks because they run as a "job" and don't have to respond to individual characters. They work the same whether they are transferring by block, by line, or by character. However, for other transfers such as system interaction with the keyboard, console displays, etc. buffering is unworkable. Each character must be sent immedaitely without waiting for an entire block or line. I shall suggest three possible solutions to this conflict.

(1.) Each program that requires intimate, immediate inter-

(Continued)

This month's cover

Artist Kim Behm chose "social realism" as a style to portray the two Altair computers from MITS in the painting reproduced on this cover of *Interface*. The reason for this choice was to graphically point out that Altair computers, because of their availability and low cost, are really "people's" computers as opposed to expensive computers designed for large corporations and universities. While hundreds of Altair computers *have been sold* to government, educational and industrial institutions, the dramatic impact of Altair has been felt most in the consumer market place. Thousands of computer hobbyists now own Altair computers and their usefulness to individuals is just now being realized. The Altair 8800 has been interfaced to most traditional computer peripherals such as CRT terminals, line printers, teletypewriters, paper tape readers, and floppy discs. It has also been interfaced to unusual devices such as the Cyclops Camera and T.V. Dazzler and many hobbyists are working on bazaar interfaces such as bio-feedback machines, model trains and planes, voice input and output, home entertainment systems, and custom burglar alarm systems.

(Continued from Page 9)

action with some device can be specifically programmed for that device instead of using the standard I/O call. Examples would be the system level keyboard input and console displays. A given program could of course use both specific and standardized I/O. An example would be an editor, calling general I/O routines for input and output of text, but permanently programmed to use a specific keyboard for text modifications and control. This is the solution used by most minicomputer operating systems.

(2.) Both types of standardized I/O could be provided in the system, both buffered and immediate (byte-at-a-time). Both types could be assignable by the system to connect to any compatible device. Most non-buffer devices, TTY, CRT, could be operated from either type of I/O call, buffered or not, whereas buffer devices, disk, tape, could only be called as buffered I/O, and console lights (for example) could only be called by a non-buffered call. It's not clear that the extra complications of maintaining dual device assignment tables would be worth the gains possible with this solution.

(3.) A compromise is possible between solutions 1. and 2. wherein the system provides standardized, buffered I/O as well as a few specific calls for specific non-buffered devices. Both types of calls would be more or less standardized (but distinct) from the point of view of a program running under the system. From the point of view of the system, buffered calls would be completely general, assignable, and easily set up for any configuration, while non-buffered character I/O would be written into the system for a particular set of devices. A program could use a different non-buffered device than it was first written for after reassembly with a different device number or pointer. Various degrees of generality could be retained here, but the overall system would be greatly simplified if these non-buffered I/O calls were not reassignable by the system.

Standard Device Handlers

The elegant generality of standardized, system controlled I/O is achieved by fitting specific device handler modules into uniform slots in the operating system structure. These device handlers are all called by a uniform subroutine call with a standard argument structure. Ideally, the device handlers present in a given system would not be coded on the same ROM with the rest of the system, but would be on separate interchangeable ROMs. Since all devic handler modules are alike as seen by the system, a new device can be added into the operating system simply by plugging in a new device handler ROM. It is not clear yet to what extent this ideal can be met in reality. One of the severe problems is that a certain handler size must be allocated in the early stages of system design. In order to expand the system capability to an unforeseen device, it must be possible to write a handler for that device which fits the previously specified handler specs. This could turn out to be a real pain, for example, for a floppy disk with out bit assembly hardware, where the I/O handler had to include the bit packing code as well as sector and track searching and perhaps handling interrupts from the drive. Of course, some handlers are simple and it's a waste of space to allocate too much room for the handler module. Keep in mind that some programs may want to use three or four (possibly different) handlers simultaneously.

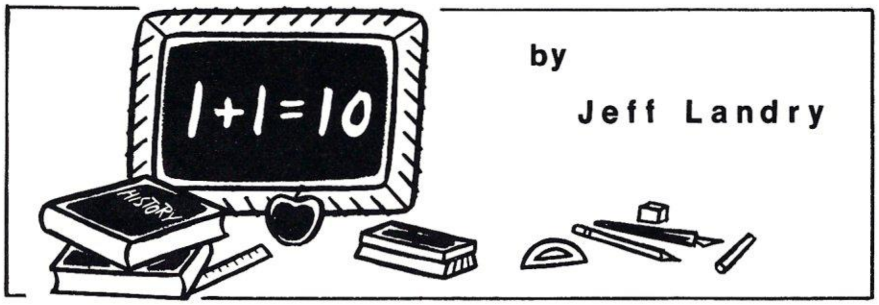
We now have a picture of a device handler module from the point of view of operating system layout. What does it look like to the calling program and to the system's device assignment routines?

We begin by describing the argument list. For a general device call, you need to specify the memory location at which the transfer starts, the device address affected by the transfer and the number of device blocks to be transferred. A bit in one of the argument words would specify a read vs. a write operation. One thing we have done without prior notice was to assume a standard system block structure for all devices. Since some devices must transfer a certain amount once they start (like a disk which writes an entire sector), the system must assume that the same minimum amount of data may be transferred any time you call a device. The difficulty here is that this decision is fixed by the requirements of some particular device and all future devices must conform to the standard. The smaller this device block is, the more efficiently storage space is used, both in memory and in device files. On the other hand, smaller blocks would require more bits in the device block address to reference a particular size device. This block address appears in many tables and arguments throughout the system where space is always at a premium. Because of the 8 bit structure of many CPU chips, the 256-word page suggests itself as the most convenient device block size. Nearly all devices, however, could transmit in smaller blocks and a page is rather large as the minimal unit of program segmentation. A block of 64 or 128 bytes is probably better. With 64-byte blocks, a 16 bit device address would allow access to any block on a device containing 4 megabytes. A device like the Teletype can of course ignore the device block address.

A standard buffered device handler call would then be something like this:

```
JMPSUBR SYSIO
        R/W DEVICEPOINTER
        DEVICEADDRESS
        MEMORYADDRESS
        BLOCKCOUNT
        program code continues
```

(Continued on Page 23)



by

Jeff Landry

Technically, the title of this article is nonsense, because a very important piece of information is missing. The lacking information is the subject of this article. Since computer programming, like mathematics, is a rigorous discipline, the terms are exact and ambiguity can cause very strange results.

The lack of precision in the title stems from the fact that in everyday living the numbers used by the people of the U.S.A. and most of the rest, if not all, of the world are based on the decimal system of numbering. A typical citizen would read the title as "one plus one equals ten" and would then recommend that the author return to the first grade to learn elementary arithmetic. Hopefully, when you have completed the reading of this article you will know that the title may also be read as "one plus one equals two", and concede that while the title is not entirely wrong, the author should certainly be more explicit in his description of numbers.

The information missing from the title is a statement of the base number of the numbering system to which the numbers in the equation belong. The concept of number systems and their base numbers is an extremely basic, but not entirely elementary, concept of everyday mathematics. Many people would consider that sufficient reason for discussing number systems, but we have a more specific rationale for understanding number systems.

That rationale is that we wish to be capable of communicating with computers with as much understanding as possible. Since humans primarily think in decimal (any numbers used in this article should be considered decimal unless otherwise stated) and computers in binary and computer texts and reference manuals use octal and hexadecimal extensively, it is fundamental for the enjoyment of one's own computer that he have as firm a grasp as possible of the concept of number systems.

Number systems all have one thing in common—they express exact quantities. Ten things are ten things regardless of whether the number used to describe them is A_{16}, 10_{10}, 12_8, or 1010_2. This basic property of numbers is only half of the reason one must be able to deal with non-decimal numbers to use a computer. The second reason is that not only are the numbers which computers crunch binary, but the instructions are coded in binary also.

To deal with the information which a computer can manipulate, one must be capable, not only of translating the decimal numbers of accounting and engineering into the binary of the computer, but, to instruct the computer in what to do with this information, the programmer must be able to correlate the binary code found in the computer with actions which the computer is capable of taking. It may not be obvious from this description, but the point which needs to be understood is that being capable of converting numbers from binary to decimal is not sufficient for a competent programmer. The programmer must be able to intimately associate the binary numbers found in the computer's memory not only with their absolute values, but also with the instruction set of the computer being used.

This article will not directly address itself to introducing both the concept of number system construction and number conversion between systems. More specifically, it will discuss the relation of binary, octal and hexadecimal to computer instruction sets.

For the quickest introduction to the construction of number systems, we will make the reasonable assumption that everyone understands the general concepts of high school arithmetic, and introduce—just to be safe—a couple of basic concepts which may have slipped the reader's mind and then analyze the construction of a decimal number. Starting with the fundamentals derived from examining the decimal system, we will construct binary, octal and hex (hexadecimal) and show how very easy it is to get from any one of these last three to any of the other three.

DEFINITIONS:

Exponents are numbers which indicate how many times a number should be multiplied by itself. Exponents are written as superscripts.

Example: $10^2 = 10 \times 10 = 100$. In this case, 2 is the exponent of 10.

$3^4 = 3 \times 3 \times 3 \times 3 = 81$. This time 4 is the exponent of 3.

Exponents are also referred to as the "power" to which a number is raised.

Example: 10^2 = 10 to the second power.

3^4 = 3 to the fourth power.

Any number to the zeroth power (n^0) is arbitrarily set equal to 1 ($10^0 = 2^0 = 8^0 — 16^0 — 1$). — 1).

The *base number* of a number system is the number which describes how many elements

(Continued)

Continued from Page 11

there are in the number system. The base number is written as a subscript (1_{10} = one base 10) and is *always* written as a decimal number (this is an absolute necessity since the base number is always "10" when written in its own system's notation).

Example: 0 and 1 are two elements which may be used as the basis of a number system. If only these two elements are used, then the base number of the system is "2". This number system is the one used for computer arithmetic and logic and is called the binary system.

If 0, 1, 2, 3, 4, 5, 6, 7 are used as the only elements of a number system, then the base of this number system is the number 8 because the system has only eight symbols or elements. This system is the octal system, loved and made so well known by, among others, the Digital Equipment Corporation's computer people.

If one were to add a zero element to the alphabet and then associate "A" with decimal "1", "B" with "2", and so on until "Z" with "26", he would have a base 27 number system. This is an extreme example of a case which occurs in computer-oriented literature, the specific case is hexadecimal, which is associated with IBM in much the same way octal is with D.E.C. Hexadecimal is a number system which requires more elements than there are special numeric characters. Hex is the base 16 system and requires 16 elements; 0-9 provides the first ten and A, B, C, D, E, and F are normally taken to serve as the eleventh, twelfth, thirteenth, fourteenth, fifteenth, and sixteenth, respectively.

NUMBER SYSTEM:

This section is intended to give a basic knowledge of how to take a number written in one number system and convert it to another number system. There are essentially two ways of doing this: 1) understand how both systems are constructed, and 2) use a table. The first method allows you to work with minimal reference to literature and should you be stuck without the tables, the job can still be done. Tables, on the other hand, have the delightful advantage of supplying answers with a minimal amount of distraction from the task at hand.

While both methods have advantages, it is necessary to point out that there are times when both are employed in an interwoven fashion due to the fact that constructing a number takes a certain amount of time and concentration and finding a complete set of tables is not always easy. Therefore, the first method will be discussed and the second utilized.

THE BASIC ELEMENTS OF FOUR NUMBER SYSTEMS

BINARY	OCTAL	DECIMAL	HEX
0	0	0	0
1	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
	6	6	6
	7	7	7
		8	8
		9	9
			A
			B
			C
			D
			E
			F

In decimal it is (hopefully!) known that each position of a number has a value which is directly related to how far to the left of the decimal point it is (also how far to the right, but this article will not concern itself with fractional numbers). To provide for a more convenient description we will ignore the fact that numbers can occur to the right of the decimal and therefore, forget the decimal point. This leaves only whole numbers, or the "integer" numbers, a term not often seen in computer instruction manuals. Thus positions will be referenced to the rightmost numeral of a number. The rightmost position is the zero position. In order to get a feeling for the concept "value of position", examine what happens when the number to be counted exceeds the quantity of numeric symbols available to a number system.

$1{,}000{,}000_{10}$ = one million

Rightmost, or "zero" position
First, or "1" position
Second, or "2" position
Sixth, or "6" position

If there are "xxxxxxxxxxx" X's to count and only ten symbols, as there are in the decimal system, one obvious way to tell others how many X's there are is to say that there are nine X's and two additional X's. It is not hard to see that a new symbol for these many things would be a definite asset; however, if it is necessary to create a new symbol every time one counts a bigger quantity of things, it wouldn't be very long before there would be more symbols than anyone could remember. To counter this possibility some bright fellow thought up the "value of position" concept. When applied, it results in a very simple set of rules:

1. Pick a specific limited set symbols, and express all numbers by using these symbols serially strung out one after the other.
2. One of these symbols must be considered to have no intrinsic value of its own, the others must all have unique intrinsic values, the smallest of which is "one", the greatest of which is the number which is equal to one, less than the total number of symbols.
3. The value of a symbol's position is determined by the system's base number and the number of the posi-

tion, and is equal to, if *n* is the position number, the base number to the nth power.

4. The number is represented by a symbol in a specific position is equal to the product of the intrinsic value of the symbol and value of the position.
5. Finally, the number represented by the string of symbols is the number which is the sum of all the numbers which each element, in its specific position of the string, represents.

What this set of rules gives us is the ability to write a specific symbol for any number we come across. Another asset not to be overlooked is that the choice and number of symbols is left up to the user of the method, and getting down to brass tacks, multiple system interpretation of numbers becomes possible with only a comparative list of the basic symbols of the involved systems. One need only, specify the zero (intrinsic value) symbol and the intrinsic value of the other symbols, and anybody who understands the rules can readily zip (?) back and forth between systems.

The disection of 1409_{10} should provide a practical feel for the use of the rules. The subscript 10 conveys the base number and the table gives the symbols whose intrinsic values, it is assumed, everyone knows.

1. Nine occupies the "zero" position, zero occupies the "one" position, four occupies the "two" position, and one occupies the "three" position.
2. In the decimal system the "0" position equals 10^0 or 1; "1" position equals 10^1 or 10; "2" position equals 10^2 or 100; "3" position equals 10^3 or 1000.
3. $1409_{10} = (1 \times 10^3) + (4 \times 10^2) + (0 \times 10^1) + (9 \times 10^0)$
   $= (1 \times 1000) + (4 \times 100) + (0 \times 10) + (9 \times 1)$

If this sounds like circular reasoning, don't get worried because it is. Without changing number systems it is impossible to get a different symbol to represent a single number. Still, the example brings out a few points of worth. Notice, for instance, that the zero adds nothing of itself to the final answer. Its value is that of giving its neighbor to the left the value of a higher position without requiring that the intrinsic value of the position which the zero fills contribute to the specified number.

Another point is that knowing the position which the most significant (left most) numeral holds tells the least value a number with that many digits may have (i.e., a four-digit decimal number may be no less than 10^3 and no greater than $10^4 - 1$). Also please note that finding the values of the individual digits in *any* order and summing them together results in the required number.

4 NUMBER SYSTEM TRANSLATION TABLE

BINARY	OCTAL	DECIMAL	HEX
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11
10010	22	18	12
10011	23	19	13
10100	24	20	14
10101	25	21	15
10110	26	22	16
10111	27	23	17
11000	30	24	18
11001	31	25	19
11010	32	26	1A
11011	33	27	1B
11100	34	28	1C
11101	35	29	1D
11110	36	30	1E
11111	37	31	1F
ETC.	ETC.	ETC.	ETC.

16 DIGIT BINARY NUMBER POSITION VALUE TABLE

POSITION	7	6	5	4	3	2	1	0
DIGIT	*	*	*	*	*	*	*	*

POSITION	POSITION VALUE (BASE 10)		
0	2^0	=	1
1	2^1	=	2
2	2^2	=	4
3	2^3	=	8
4	2^4	=	16
5	2^5	=	32
6	2^6	=	64
7	2^7	=	128
8	2^8	=	256
09	2^9	=	512
10	2^{10}	=	1024
11	2^{11}	=	2048
12	2^{12}	=	4096
13	2^{13}	=	8192
14	2^{14}	=	16384
15	2^{15}	=	32768

To test the validity of what has been said, it is necessary to show a few actual conversions from one system to another. The simplest (and most pertinent to our interest in computers) number system is binary. The set of elements consists of only two, as shown in the first table. This makes things very easy because a numeral in a position is either the value of the position or has no value, requiring us to do no multiplication. We need only find the position value, using the provided table, of each position with a "one" in it and add them together to get a number which means something (the author is assuming the decimal numbers convey more meaning to the reader than any other system would; if this is not the case, then the position

(Continued on Page 20)

THE APPLICATIONS EXCHANGE

EDITOR'S NOTE: Beginning with this issue, SCCS INTERFACE will present the following column and Larry Press will provide an invaluable service which futhers one of the principle goals of this publication.

If you finally got your ALTAIR up and running and now you can't think of anything to do with it but watch the lights blink or use it for a fly swatter, then we invite you to be a reader of this column.

On the other hand, if you are doing something with your baby or at least you have some ideas as to what you would do if you only had one, then you could be a contributor to this column.

We plan to **edit** the column, not **write** it. Therefore, this is a request for contributions. We would like to print external descriptions of applications and programs that you either have up or have dreamed up. If space and interest permit, internal details will also be presented. We'll also maintain a library of **all** the details and supporting materials you might contribute, i.e., (listings, tapes, documentation, schematics, relevant references, etc.) regardless of how much detail is printed in INTERFACE. That way people who wish more information can either contact us or, better yet, they can contact the original contributor.

The ideas don't have to be your own and they may be elaborate or simple, short or long. Whatever you feel might be of interest to someone else. You are also encouraged to submit relevant articles that you see elsewhere, which we will edit and reterence.

We hope that you will take the time to share your ideas for applications with the rest of us.

WRITE: Larry Press,
Applications Exchange Editor
128 Park Place.
Venice, Califrnia 90291

(Continued on Page 17)

ALTAIR SWITCH MEMORY

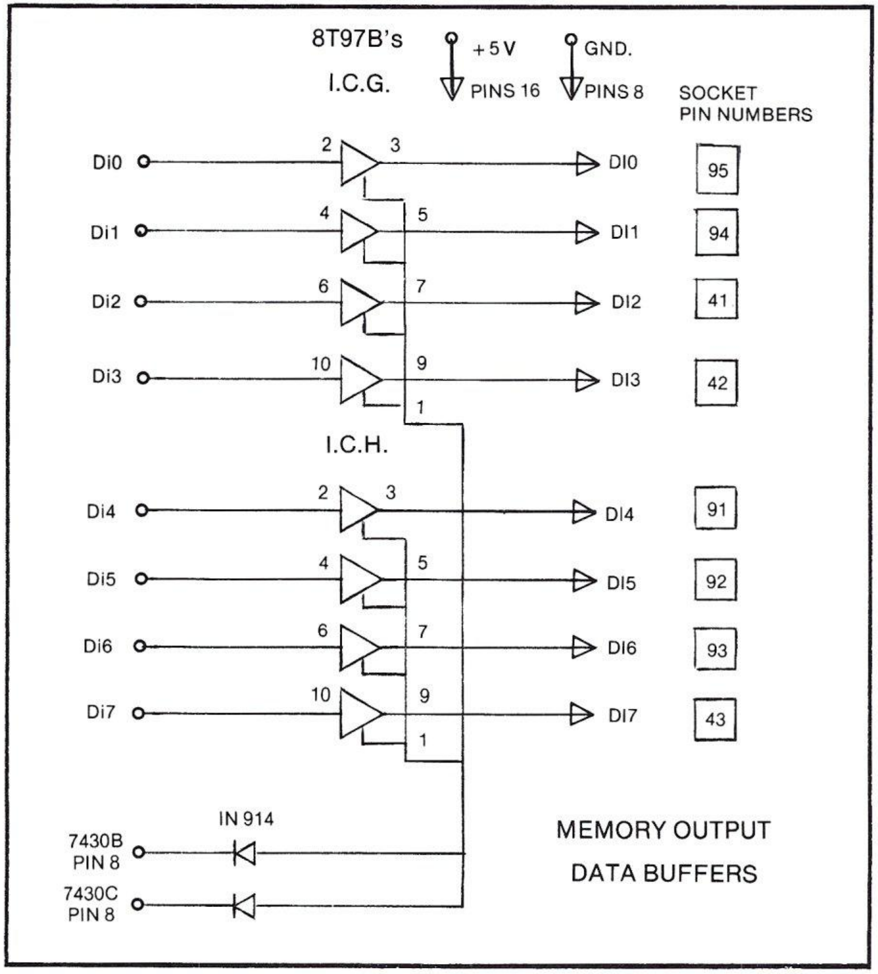
S. Wilcox

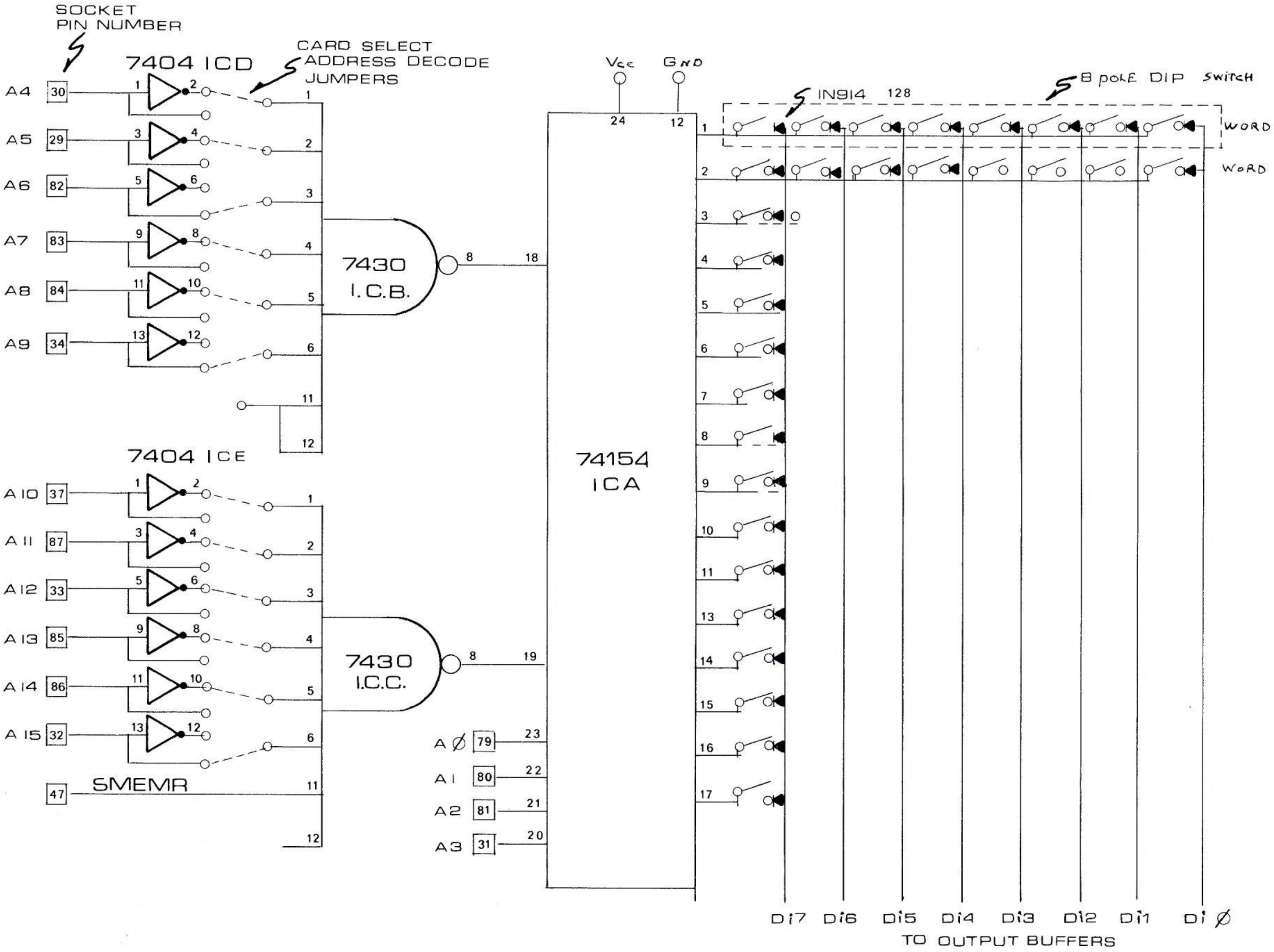
One thing it seems any computer system can never have enough of is memory whether it be RAM, ROM, PROM, EROM, RMM, etc. Semiconductor memory prices are dropping like bricks, but even so here is a different type of memory system that can prove convenient. It is a manually programmable and eraseable memory made from diodes and DIP switches, you know those multiple jobs that are the same size as duel-in-line plastic I.C.'s. This makes for a rugged memory that itself is unaffected by power loss, clocking refresh, or noise to any great extent.

This type memory could be made in any size you wanted; this one is only sixteen 8 bit words. Sixteen words you say is too little to be worthwhile bothering about, well not if you use the diode matrix memory to hold bootstrap loaders or interrupt service routines. (Incidentally this could be hard wired without any DIP switches and reduce cost considerably.)

The memory board which I built is on a prototype board and contains all circuitry necessary to just plug in to the ALTAIR. On board decoding is provided as well as output tri-state buffering. The output buffers are not really necessary if you select diodes with a very low forward resistance and a very high reverse resistance and you do not have a terminated data bus such as provided by Processor Technology on their Mother Board. Also no processor slow down "wait" circuitry is necessary.

I use this setup to load a tape input program from a cassette which in turn loads the rest of the RAM memory. For those who cannot as yet afford any EPROM or a programmer for them and want a small amount of read only memory to check out short routines, this would seem to fill the bill.




SOCKET PIN NUMBER
7404 ICD
CARD SELECT ADDRESS DECODE JUMPERS
A4 30
A5 29
A6 82
A7 83
A8 84
A9 34
7430 I.C.B.
7404 ICE
A10 37
A11 87
A12 33
A13 85
A14 86
A15 32
47 SMEMR
7430 I.C.C.
Vcc
GND
24
12
18
19
74154 ICA
A0 79 23
A1 80 22
A2 81 21
A3 31 20
IN914 128
8 pole DIP switch
WORD
WORD
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
TO OUTPUT BUFFERS

SWITCH & DIODE MEMORY MATRIX

ARROWHEAD COMPUTER CO.
The
COMPUTER STORE
11656
OPEN

World's First

Now in Los Angeles . . .

ALTAIR 8800 Computers

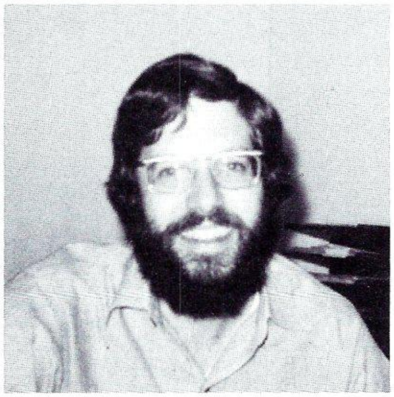
Kits in stock
- CPU's
- Memories
- I/O Interfaces

Assembled units by order
Assembly advice
Configuration planning
PLUS Books on computing

ALTAIR BASIC Manual @ $5.00
(Fantastic, unique software!)
8800 Hardware Manual @ $15.00

"THE COMPUTER STORE"

Arrowhead Computer Co.
11656 W. Pico Blvd.
Los Angeles, Ca. 90064
(213) 478-3168
(¼ mile west of the
San Diego Freeway,
at Barrington)
Wed. thru Fri. 2-8 pm
Sat. and Sun. 10 am - 6 pm



Dick

Lois

Special Altair®

MITS-MAS

ALTAIR 8800 COMPUTER
mits

Christmas Catalog

Lowest Price in the World!

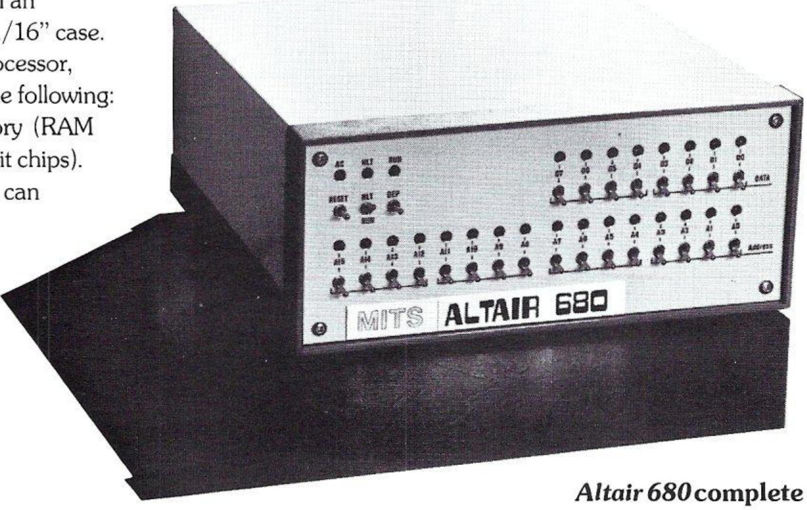
In January of 1975, MITS stunned the computer world with the announcement of the *Altair 8800* Computer that sells for $439 in kit form.

Today MITS is announcing the Altair 680.

The Altair 680, built around the revolutionary new 6800 microprocessor chip, is the lowest priced complete computer on the market. Until December 31, 1975, this computer will be sold in kit form for the amazing introductory price of $293! **(A savings of $52!)**

The Altair 680 comes with power supply, front panel control board, and CPU board inclosed in an 11" wide x 11" deep x 4 11/16" case. In addition to the 6800 processor, the CPU board contains the following:

1. 1024 words of memory (RAM 2102 type 1024 x 1-bit chips).
2. Built-in Interface that can be configured for RS232 or 20 mA Teletype loop or 60 mA Teletype.
3. Provisions for 1024 words of ROM or PROM.



The Altair 680 can be programmed from the front panel switches or it can be connected to a computer terminal (RS232) or a Teletype such as an ASR-33 or surplus five-level Baudott Teletype (under $100).

The Altair 680 can be utilized for many home, commercial or industrial applications or it can be used as a development system for *Altair 680* CPU boards. With a cycle time of 4 microseconds, 16-bit addressing, and the capability of directly addressing 65,000 words of memory and a virtually unlimited number of I/O devices, the *Altair 680* is a very versatile computer!

Altair 680 Software

Software for the *Altair 680* includes a monitor on PROM, assembler, debug, and editor. This software is available to *Altair 680* owners at a nominal cost.

Future software development will be influenced by customer demand and *may include BASIC on ROM.* MITS will sponsor lucrative software contests to encourage the rapid growth of the *Altair 680* software library. Programs in this library will be made available to all *Altair 680* owners at the cost of printing and mailing.

Contact factory for updated information and prices.

Altair Users Group

All *Altair 680* purchasers will receive a free one year membership to the Altair Users Group. This group is the largest of its kind in the world and includes thousands of *Altair 8800* and *680* users.

Members of the Altair Users Group are kept abreast of Altair developments through the monthly publication, ***Computer Notes.***

Altair 680 Documentation

The *Altair 680* kit comes with complete documentation including assembly manual, assembly hints manual, operation manual, and theory manual. Assembled units come with operation and theory manuals. Turnkey model and CPU boards also include documentation.

NOTE: *Altair 680 manuals can be purchased separately. See back page of this catalog for prices.*

Delivery

Personal checks take 2-3 weeks to process while money orders and credit card purchases can be processed in 1-3 days. Delivery should be 30-60 days but this can vary according to order backlog. All orders are handled on a first come, first served basis.

Altair 680 Prices

***Altair 680* complete computer kit** **($345 after December 31, 1975)**	**$293**
***Altair 680* assembled and tested**	**$420**
***Altair 680T* turnkey model (complete *Altair 680* except front panel control board) Kit Only** **($280 after December 31, 1975)**	**$240**
***Altair 680* CPU board (including pc board, 6800 microprocessor chip, 1024 word memory, 3 way interface and all remaining components except power supply)** **($195 after December 31, 1975)**	**$180**
***Altair 680* CPU board assembled and tested**	**$275**
Option I/O socket kit (required when interfacing 680 to external devices)	**$ 29**
Option cooling fan (required when expanding 680 internally) **($22 after December 31, 1975)**	**$ 16**
Option cooling fan installed	**$ 26**
PROM kit (256 x 8-bit ultraviolet, erasable 1702 devices)	**$ 42**

MITS
"Creative Electronics"

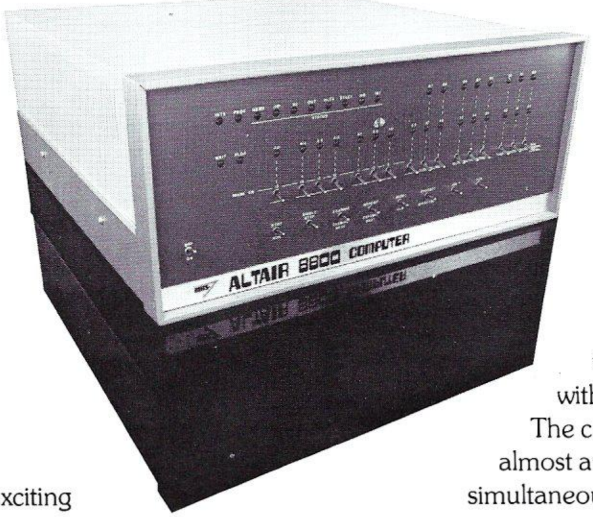
Prices, delivery and specifications subject to change.

"PROJECT BREAKTHROUGH!
World's First Minicomputer Kit To Rival Commercial Models . . . 'Altair 8800' "

headline on cover of *Popular Electronics*, January, 1975

The Altair 8800 from MITS is now one of the most successful computers ever delivered. Thousands of *Altair 8800's* have been sold and are in the field where they are being used for an infinite variety of industrial, business, science and home applications.

The *Altair 8800* is extensively supported by ongoing hardware and software development. *Altair 8800* interface and memory modules and *Altair peripherals* are inexpensively priced, yet among the highest quality in the business. Byte for byte, *Altair 8800* BASIC language software is the most powerful BASIC ever written.



Thanks to the success of the Altair 8800, building and programming computers has become one of the World's most exciting and fastest growing hobbies. Local Altair 8800 Users Clubs have been formed across the United States and in such far away places as England and Japan.

Thanks to clean, efficient design and accurate, easy to understand assembly instructions, the *Altair 8800* is an easy kit to assemble. As an *Altair 8800* kit builder, you will have the satisfaction of successfully building your own computer and you will learn about the internal structure of digital computers.

As the owner of an Altair 8800, you will be backed by the technical expertise of the MITS Customer Service Department. You will receive the latest update information, programming hints, technical advice and general computer information on a monthly basis through a free subscription to *Computer Notes.* You will be in contact with other *Altair 8800* owners through the *Altair Users Group* and you will have access to the extensive *Altair 8800 Software Library.*

No other computer on today's market can offer you as much support as the Altair 8800.

Christmas Special

For a limited time only, you can be the owner of an *Altair 8800* with a 1,024 word memory module for just $68 a month! See back page of this catalog for all the details.

Altair 8800 Features

Built around the most successful (and many say the most powerful) microprocessor chip ever [the Intel 8080], the *Altair 8800* is a variable word length computer with an 8-bit processor, 16-bit addressing and a maximum word size of 24-bits. It has 78 basic machine instructions with variances over 200 instructions. The *Altair 8800* can directly address 256 input and 256 output devices and up to 65,000 words of memory.

Up to 300 peripherals can be interfaced to the *Altair 8800* without any additional buffering. The custom designer can interface almost any number of imaginable devices simultaneously. All *Altair* peripherals are supplied with software handlers to make interfacing easy.

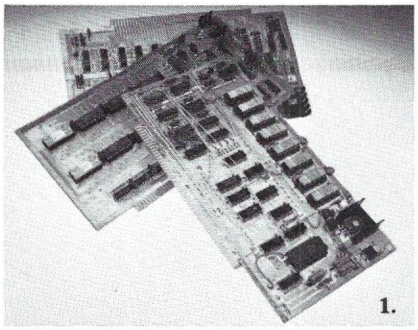
The *Altair 8800* includes the CPU board, front panel control board, power supply (enough to power any additional cards), and expander board (with room for 3 extra interface or memory modules) all inclosed in a handsome, aluminum case complete with sub-panel and dress panel. Up to 16 cards can be added inside the main case.

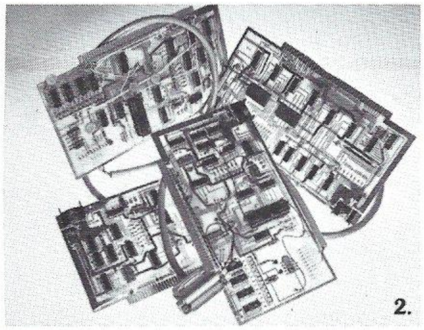
Altair 8800 Prices

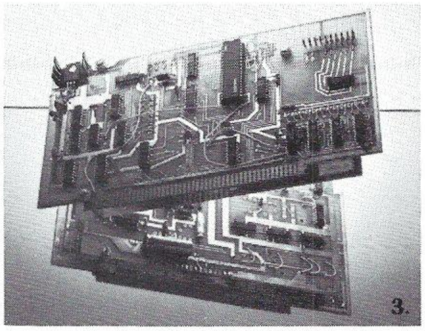
Altair 8800 Computer kit (includes assembly hints, assembly, operator's, and theory manuals)	**$439**
Altair 8800 assembled	**$621**
Expander board (adds 4 slots)	**$ 16 kit**
assembled,	**$ 31**
Cooling fan	**$ 16 kit**
assembled,	**$ 20**

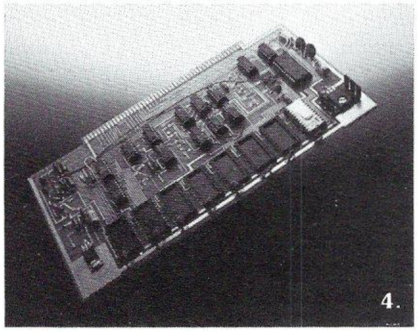
Altair Modules –

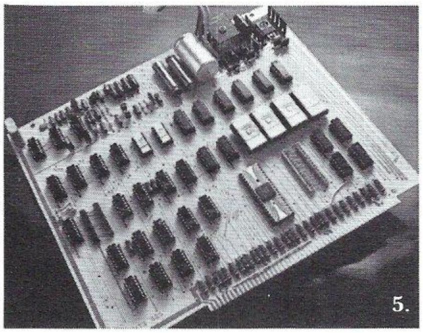
Custom Design Your Computer System to Meet Your Application *(and stay within your budget!)*

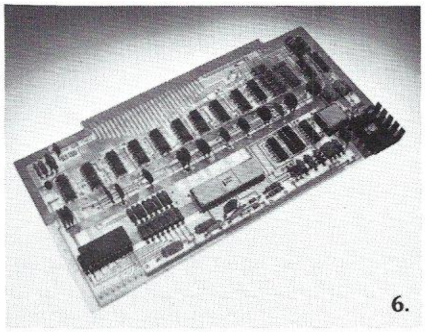

1.


2.


3.


4.


5.


6.

1. Three 8800 memory modules. Three memory modules are now available for the *Altair 8800* and more are in the works. These modules include a 1K (1,024) word static card, a 2K (2,048) word static card, and a 4K (4,096) word dynamic card. Each of these modules is constructed with the finest components available and each contains memory protect features (prevents the computer from accidently writing over programs you want to save). The maximum access time of the static cards is 850 nanoseconds while access time for the dynamic card is 300 nanoseconds.

2. Four 8800 Interface Modules. Interface modules now available for the *Altair 8800* include three serial and one parallell I/O cards. The SIOA serial card is used to connect the *Altair 8800* to CRT's and other computer terminals that have industry standard RS232 asynchronous interconnect lines. It has divider logic to allow for presettable baud rates up to 19,200 baud (5-8 data bits).

The SIOB is the same as the SIOA except that all signals are TTL levels. This card is a general purpose serial interface that can be used to custom interface the *Altair 8800* to a wide variety of devices.

The SIOC is also the same as the other serial cards except it is used to interface the *Altair 8800* to conventional Teletypes and other asynchronous 20 mA current loop terminals. The SIOC is required to interface the *Altair* with ASR-33 Teletypes available from MITS.

The PIO parallel interface card is used for bidirectional transmission of bytes at speeds up to 25,000 bytes/second! It is a full TTL compatible input/output card with necessary handshake flags for conventional parallel interface. Both input and output data have their own 8-bit latch for buffering. Includes necessary logic to allow an adjacent channel to be a control channel. Most commonly used to interface the *Altair 8800* to SWTPC-TVT's or equivalent, custom A/D-D/A interfacing, computer to computer interfacing, and control applications.

3. Audio-Cassette Interface. This best-selling *Altair* module allows you to connect your *Altair 8800* to any tape recorder (medium quality cassette is adequate) for inexpensive mass storage. It works by modulating (changing) digital signals from the computer to audio signals for recording data and by demodulating the audio signal for playback. Consists of a special *Altair* modem board "piggy-backed" on an SIOB board. Requires one slot in the 8800.

4. 8800 PROM module. This PROM memory card is designed to hold up to 2K of PROM. Contact factory for price and other information. *Altair 8800* PROM Programmer to be announced soon.

5. 680 CPU board. The *Altair 680* CPU board is a complete computer on a board (less power supply). In addition to the 6800 CPU available from Motorola and AMI, the *Altair 680* CPU board comes with 1K of RAM memory (2102 type, 1024 x 1-bit chips), built-in I/O that can be configured for RS232 or 20 mA current loop or 60 mA current loop, and provisions for 1K of ROM or PROM. It measures 8¾" x 10¼".

6. 8800 CPU board. The heart of the *Altair 8800* is its CPU board. This double-sided board was designed around the powerful byte oriented, variable word length 8080 processor—a complete central processing unit on a single LSI chip using n-channel, silicon gate MOS technology. The CPU board also contains the *Altair* system clock—a standard TTL oscillator with a 2.000 MHz crystal as the feedback element.

Altair Module Prices:

1K static memory	**$97 kit** and **$139 assembled**
2K static memory	**$145 kit** and **$195 assembled**
4K dynamic memory	**$195 kit** and **$275 assembled**
SIOA interface	**$119 kit** and **$138 assembled**
SIOB and SIOC interface	**$124 kit** and **$146 assembled**
PIO interface	**$92 kit** and **$114 assembled**
Audio-cassette interface	**$128 kit** and **$174 assembled**
680 CPU board kit	**$180 kit**
[$195 after December 31, 1975]	
680 CPU board, assembled	**$275**
8800 CPU board	**$310 kit** and **$360 assembled**

NOTE: ***Watch our advertisements for announcement of new Altair 8800 modules and Altair 680 modules.***

BASIC language was chosen for the Altair 8800 because it is the *easiest language to learn* and because it can be used for an *infinite number of applications*. Literally hundreds of thousands of BASIC programs have been written and are in the public domain. These programs include accounting programs, business programs, scientific programs, educational programs, game programs, engineering programs, and much more.

Altair BASIC is an *interactive language*. This means that you get immediate answers and you can use your Altair as a super programmable calculator as well as for writing complicated programs.

> *“I've seen and used other BASICs, but byte-for-byte, Altair is the most powerful BASIC I've seen. I'm particularly impressed with the n-dimensional arrays (and for strings too!), machine level I/O, and machine language 'function' features. The level of your documentation is, for me, though the high point. Sections for those who know nothing and sections for those who know a lot, plus sections that 'normal' people can read and understand.”*
>
> **J. Scott Williams**
> **Bellingham, Washington**

Altair BASIC was written as efficiently as possible to allow for the *maximum number of features in the minimum amount of memory.* You can order one of three Altair BASICs: 4K BASIC-designed to run in an Altair 8800 with as little as 4K of memory, 8K BASIC, or EXTENDED BASIC (12K). Each of these BASICs allows you to have *multiple statements per line* (a memory saving feature), and each of them is capable of executing *700 floating point additions* per second!

The 8K BASIC and EXTENDED BASIC have *multi-dimensioned arrays* for both strings and numbers. This is particularly useful for applications requiring lists of names or numbers such as accounting programs, inventory programs, mailing lists, etc.

The 8K BASIC and EXTENDED BASIC also have an OUT and corresponding INP statement that allows you to use your Altair 8800 *control low speed devices* such as drill presses, lathes, stepping motors, model trains, model airplanes, alarms, heating systems, home entertainment systems, etc.

Altair BASIC comes with complete documentation including a copy of *"My Computer Likes Me When I Speak in BASIC"* by Bob Albrecht, a beginner's BASIC text.

Never before has such a powerful BASIC language been marketed at such low prices!

4K BASIC Features

Altair 4K BASIC leaves apporimxately 750 bytes in a 4K Altair for programming which can be increased by deleting the math functions. This powerful BASIC has **16 statements** [IF . . . THEN, GOTO, GOSUB, RETURN, FOR, NEXT, READ, INPUT, END, DATA, LET, DIM, REM, RESTOR, PRINT, and STOP] in addition to 4 commands [LIST, RUN, CLEAR, SCRATCH] and **6 functions** [RND, SQR, SIN, ABS, INT and SGN]. Other features include: *direct execution* of any statement except INPUT; an "@ " *symbol* that deletes a whole line and a "←" that deletes the last character; *two-character error code* and line number printed when error occurs; *Control C* which is used to interrupt a program; *maximum line number of 65, 535;* and all results calculated to at least six *decimal digits of precision.*

8K BASIC Features

Altair 8K BASIC leaves approximately 2K bytes in an 8K Altair for programming which can also be increased by deleting the math functions. This BASIC is the same as the 4K BASIC only with **4 additional statements** [ON . . . GOTO, ON . . . GOSUB, OUT, DEF], **1 additional command** [CONT] and **8 additional functions** [COS, LOG, EXP, TAN, ATN, INP, FRE, POS]. Other additional features include *multi-dimensioned arrays* for both strings and numbers, *AND, OR, NOT* operators that can be used in IF statements or forumlas, *strings* with a maximum length of 255 characters, *string concatenation (A$ = B$)* and the following string functions: *LEN, ASC, CHAR$, RIGHT$, LEFT$, MID$, STR$, and VAL.*

EXTENDED BASIC

Altair EXTENDED BASIC is the same as 8K BASIC with the addition of *double precision arithmetic, PRINT USING* and *disk file I/O.* A minimum of 12K memory is required to support EXTENDED BASIC.

Other Altair 8800 software includes a Disk Operating System, assembler, text editor, and system monitor. Altair users also have access to the *Altair Library,* which contains a large number of useful programs.

SOFTWARE PRICES:

Altair 4K BASIC	**$150**
Purchasers of an Altair 8800, 4K of Altair memory, and an Altair I/O board	**$ 60**
Altair 8K BASIC	**$200**
Purchasers of an Altair 8800, 8K of Altair memory, and an Altair I/O board	**$ 75**
Altair Extended BASIC	**$350**
Purchasers of an Altair 8800, 12K of Altair memory, and an Altair I/O Board	**$150**
Altair PACKAGE ONE (assembler, text editor, system monitor)	**$175**
Purchasers of an Altair 8800, 8K of Altair memory, and an Altair I/O board	**$ 30**
Altair Disk Operating System	**$500**
Purchasers of an Altair 8800, 12K of Altair memory, Altair I/O and Altair Floppy Disk	**$150**

Note: ***When ordering software, specify paper tape or cassette tape.***

Inexpensive, Sophisticated Mass Storage

ALTAIR DISK

The *Altair Disk* can store over 300,000 words of information on a floppy disk! It offers the advantage of nonvolatile memory (doesn't "forget" when power is turned off) and fast access to data (¾ seconds—worst case). The data transfer rate of the *Altair Disk* to and from the computer is a whopping 250,000 bits per second.

The *Altair Disk* includes the disk controller, disk drive, one floppy disk and a software driver. The disk controller, which consists of two cards requiring two slots in the 8800, is capable of controlling up to 16 disk drives. It controls all mechanical functions of the disk, presents the disk status to the computer, and converts serial data to 8-bit parallel words and vice versa for rapid transfer.

The disk drive consists of a Pertec FD400 drive mounted in an *Altair* case including power supply, a buff/multiplexer board, and cooling fan. Its rotational speed is 360 rpm, track to track access time is 10 msec., average time to read or write is 400 msec., and disk life is over 1 million passes per track.

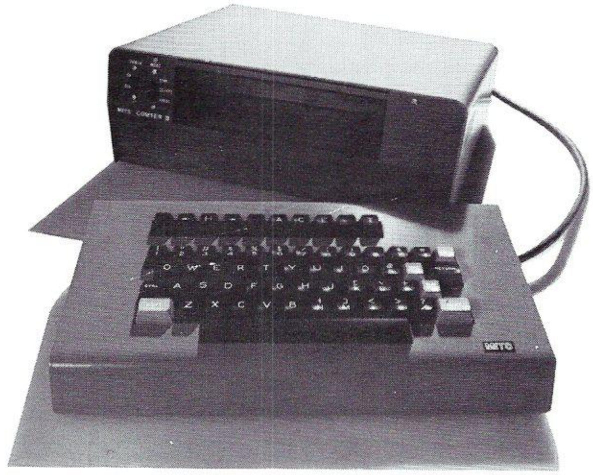
The floppy disk is hard sectored for 32 sectors per track (128 words per sector). There are 77 tracks on each disk. Extra floppy disks are available for $15 from MITS.

A Disk Operating System (software) with complete file structure and utilities for copying, deleting and sorting files is also available.

PRICES:

88-DCDD Altair Disk (includes disk controller, disk drive, one floppy disk and software driver)	**$1,480 kit**
	assembled, $1,980
88-DISK Altair Disk Drive only	**$1,180 kit**
	assembled, $1,600
Floppy disk	**$ 15**
Disk Operating System (if purchased separately)	**$ 500**
Purchasers of an Altair 8800, 12K of Altair memory, Altair I/O and Altair Disk (88-DCDD)	**$ 150**

Build Your Own Advanced Terminal!



The Comter II is easily the most advanced computer terminal kit on the market. It has its own internal memory of 256 characters which combines with a highly-readable, soft orange 32 character display to provide ease of operation and information retrieval.

Complete cursor control allows you to move data in and out of the display and operate the Comter II with the versatility of a CRT terminal. **Built-in audio-cassette interface allows you to store unlimited data from the computer and feed that information back into the computer.**

Other features include auto transmit which allows line-by-line transmission of data or program information to the computer from the Comter's memory. The Comter II has a *complete ASCII encoded keyboard* with TTY-33 format plus the addition of special function keys. Its total weight is just 15 lbs.

Flexible power requirements allow you to operate the computer at either 95-125V or 190-250V. The Comter II can be interfaced to any computer with an RS232 serial interface. Requires an SIOA board to be connected to the Altair 8800. No interface required to connect to the Altair 680.

PRICES:

Comter II kit with built-in audio cassette I/O	**$780**
Comter II assembled	**$920**

High Speed Printing at Low Cost!

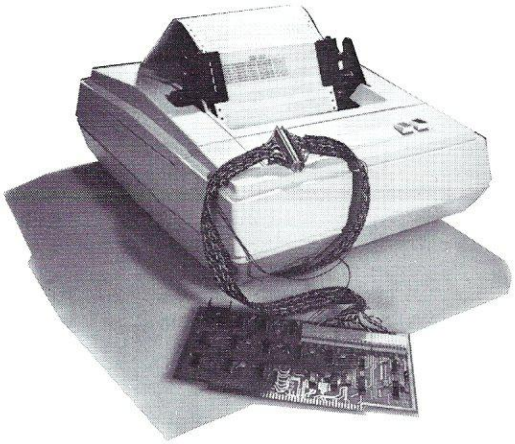
The *Altair 110 Line Printer* is a desktop line printer that produces 80 columns of 5 x 7 dot matrix characters at 100 characters per second x 70 lines per minute. The impact head prints bidirectionally on a 8½" roll paper* using a conventional teletype ribbon. The *Altair Line Printer* will print up to four copies of any item.

Maximum reliability is provided by a mechanism which contains no brakes, clutches, dampers or stepper motors. All control electronics including one-line buffer and self-test circuitry are contained on a single 5" x 15" printed circuit card. The Model 110 was expressly designed for the simplicity, reliability and extremely low cost required by current small-scale data handling systems and terminals.

Vibration and wear are minimized because the print head moves uniformly in both directions and pauses only at the end of each line. Opto-electronic sensing is used to accurately position each dot and permit characters to be printed on the fly.

The *Altair 110 Line Printer* comes with complete control electronics including a printer control card. Requires one slot in the *Altair 8800.*

**Pin-fed Optional.*



PRICES:

Altair Line Printer with
controller card .... $1,750 kit
.... assembled, $1,975

Very Low Cost Terminal

DATA IN
DATA OUT
MITS
VLCT

The *Altair VLCT* is ideal for machine language programming. It converts a three digit octal code directly into an 8 digit binary code for transmission to the computer and then displays the binary output from the computer in a 3 digital octal format. This allows you to program in octal which is much easier than programming in binary. In addition, the VLCT is much more convenient to work with than using the front panel switches of the *Altair 8800.*

PRICES:

VLCT .... $129 kit
.... assembled, $169

NOTE: ***PIO interface module is required to connect VLCT to Altair 8800.***

Teletype — *Most Versatile*

With a built-in paper tape reader and punch and hard copy output, this *ASR-33 Teletype* is perhaps the most versatile of all low cost input/output devices.

The *ASR-33 Teletype* prints 10 characters per second. It is a completely checked-out machine with standard 120 day warranty.

PRICE:

ASR-33 Teletype $1,500

NOTE: ***SIOC interface module is required to connect Teletype to Altair 8800.***

Christmas Time Payment Plan

1K Altair for Just $68 a Month!

You can be the owner of an Altair 8800 with a 1,024 word memory module for *just $68 a month.* Each month (for 8 months) you send in your payment and we send you part of an Altair kit until you have the complete system. The advantages of the plan are *NO interest* or financing charge, *GUARANTEED price* based on today's price, and *free, immediate membership to the Altair Users Group* including subscription to Computer Notes.

Our terms are cash with order, BankAmericard, or Master Charge. If you send in an early payment, we will make an early shipment. By the same token, a late payment will result in a late shipment. (After 60 days past due, the balance of the deal is cancelled. All payments must be made within 10 months).

Total $544.00 (Retail price: Altair 8800 $439.00, Memory $97.00, Postage and Handling $8.00 – total $544.00)

Altair Users Group Special

Each month the Altair Users Group sponsors a software contest and each month MITS gives away $130 in credit to the winners of this contest. At the end of the year, the author of the overall best program will receive $1000 in credit.

Membership to the Altair Users Group (the largest of its kind in the world) is free to Altair 8800 owners. However, even if you don't own an Altair, *you can be an associate member for one year at the special low price of $10 (regularly $30).*

As a member of the Altair Users Group, you will be kept informed of Altair developments, software contests, and general computer news through the monthly publication, Computer Notes. You will have access to the Altair Software Library and you can communicate to other Altair users throughout the world.

Note: These specials expire on January 30, 1976.

Altair Manuals

Altair 8800 Operators	$ 7.50
Altair 8800 Assembly	$ 9.00
Altair 8800 Theory, Schematics	$10.00
Altair 680 Operators	$ 7.50
Altair 680 Assembly	$ 7.50
Altair 680 Theory, Schematics	$10.00
BASIC Language Documentation	$10.00
Assembler, Monitor, Editor	$ 7.50
8800 4K Memory (includes Assembly, Theory & Schematics)	$ 5.00
8800 2K Memory	$ 5.00
8800 1K Memory	$ 5.00
8800 SIOA	$ 5.00
8800 SIOB	$ 5.00
8800 SIOC	$ 5.00
8800 PIO	$ 5.00
8800 ACR	$ 5.00
COMTER II Operators	$ 6.50
COMTER II Assembly	$10.00
COMTER II Theory, Schematics	$10.00
VLCT (Assembly, Operators, Theory)	$ 5.00
Altair Line Printer Interface	$ 5.00

Documentation Special One

Altair 8800 Operators, Assembly and Theory manuals plus BASIC Language manual (regularly $36.50). *Now just $15.00.*

Documentation Special Two

Altair 680 Operators, Assembly and Theory manuals. Regularly $25. *Now just $14.50.*

Documentation Special Three

BASIC Language manual, Package I (Assembler, Monitor, Editor), and BASIC language beginners text (My Computer Likes Me When I Speak BASIC by Bob Albrecht). Regularly $19.50. *Now just $12.50.*

MITS/6328 Linn NE/Albuquerque, NM 87108
505-265-7553 or 262-1951

Mail this special Altair Coupon Today!

☐ Enclosed is check for $______ ☐ BankAmericard #______ ☐ or Master Charge #______

☐ Altair 8800 ☐ Kit ☐ Assembled ☐ Fan ☐ Christmas Time Plan ☐ Expander Board

☐ Altair 680 ☐ Kit ☐ Assembled ☐ Fan ☐ I/O Sockets

☐ Teletype ☐ Line Printer ☐ Comter II ☐ VLCT ☐ Disk ☐ Disk Drive Only ☐ Disk Controller Only

☐ Memory Module ☐ I/O Module (list on separate sheet)

Postage & Handling: Add $8 for Altair 8800 or Altair 680 and $3 for any cards or peripherals if ordered separately. Teletype and Line Printer shipped by collect freight. Warranty: 90 days on parts for kits and 90 days on parts and labor for assembled units. Prices, specifications, and delivery subject to change.

☐ Documentation Special One ☐ Documentation Special Two ☐ Documentation Special Three

☐ Altair Users Group plus Computer Notes

☐ Please put me on your mailing list

88

BILL TO:

NAME______

ADDRESS______

CITY______

STATE & ZIP______

COMPANY (IF APPLICABLE)______

SHIP TO:

NAME______

ADDRESS______

CITY______

STATE & ZIP______

☐ Please send Christmas Card to above address announcing gift and anticipated delivery.

MITS/6328 Linn NE/Albuquerque, NM 87108 505-265-7553 or 262-1951

(Continued from Page 7)

least noisy — mount the power switch just below the line cord. Even if your machine is positioned in such a way that the back is easily reached, the line cord protects the power switch from accidentally being flipped, especially if mounted so that it toggles sideways rather than up and down.

What about that hole in the front existing panel? Easy. Fill it up with an LED (light emitting diode). Use the existing holes of the power switch on the front PC board, one short jumper and a 150 ohm resistor, and you will have an indicator displaying line voltage surges. As the LED becomes dimmer, it shows line voltage dropping. Air conditioning systems, toasters, etc. can pull an already low line voltage down to the point that your supplies will go out of regulation. Thus, the LED is an early warning system for your computer. When the LED extinguishes completely, there isn't enough line voltage left to produce accurate results from the computer.

HARDWARE NOTES

by S. Wilcox

ALTAIR 8800 owners who have bought or ordered the 4K memory board from Godbout Electronics should know that this is a high quality kit (sockets for all I.C.'s) but there is a pitfall.

When I received my kit and spread all the parts out on the kitchen table I was impressed by all the sockets, capacitors, and regulators. But five voltage regulators!??" Why did MITS need only one regulator with a heat sink and this memory need five without a heat sink? I was suspicious but decided to start with the assembly.

Two hours of soldering, a check of the board and it was time to plug it in and power up. I examined a memory location, checked memory protect and deposited some bit patterns. Things appeared to be fine, but the regulators are getting pretty warm. Power down. Plug in the extender card and put a milliamp meter in the +8v buss to the card; power up. Wow, on the 500ma scale to the meter pinned! Change scales and check again, it's 1.75 amps! This is obviously not a low power memory.

I next plugged in the MITS 4K board to compare and found it draws only 470ma. With the basic ALTAIR power supply rated at 8 amps for 16 plug in cards it doesn't take a mental giant to figure that is an allotment of 500ma per card. Well, this board has just eaten up the power allotment for almost four cards.

As I said, this is a quality kit and an inexpensive way to increase the memory capacity of your system, but I wouldn't recommend adding more than one of these boards without beefing up the power supply +8 volts. I would also recommend adding heat sinks to the regulators since they do seem to run warmer than any of the MITS cards do.

EDITOR'S NOTE: The difference in power consumption is that Godbout Electronics units use static memory devices and MITS units are dynamic.

* * *

(Continued from Page 14)

APPLICATION EXCHANGE

ALTAIR 8800 NOISE CURE

It seems noise is on a lot of people's minds since it is one of those things than can lay in the weeds to spring up and get you any time. The following modification suggestions were mailed anonymously to us from New Jersey.

1.) Connect buss line 72 from the D/C board to mother board line 3 not 72.

2.) Install the following capacitors on the D/C Board:

CAPAC.	FROM	TO
.001 µf	IC T pin 7	IC T pin 8
.001 µf	IC X pin 1	IC X pin 3
.001 µf	IC X pin 5	IC X pin 8 (gnd)
100 pf	IC X pin 7	ground
.001 µf	IC X pin 9	ground
.001 µf	IC X pin 11	ground
100 pf	IC X pin 13	IC X pin 15 (gnd)

3.) Install the following termination resistors to ground (Buss lines 50 and/or 100).

Note: Do not add these resistors if you have the already terminated Mother Board from Processor Technology in your system.

BUSS #	TERMIN.
21	510 ohms ¼ watt
24	510 ohms ¼ watt
25	510 ohms ¼ watt
35	510 ohms ¼ watt
36	510 ohms ¼ watt
38	510 ohms ¼ watt
39	510 ohms ¼ watt
40	510 ohms ¼ watt
49	510 ohms ¼ watt
53	1k ohms ¼ watt
68	510 ohms ¼ watt
76	510 ohms ¼ watt
77	510 ohms ¼ watt
78	510 ohms ¼ watt
88	510 ohms ¼ watt
89	510 ohms ¼ watt
90	510 ohms ¼ watt

Don't underrate noise control. An unwanted bit change due to noise can make a program go wild and self-destruct.

EDITORS NOTE: We haven't had the opportunity to check this modification so look it over carefully and let us know how it works for you. We also hope to present the manufacturer's suggestions and comments in one of our future editions.

* * *

games and things

by Mike Stern

Last issue in this column, we used a FORTRAN statement that may be unfamiliar to some people. The FORMAT statement (40) was coded as:

FORMAT (' A DESIRED SPEED OF ', and could have also been coded as:

FORMAT (20H A DESIRED SPEED OF , In our various tiptoeing through four FORTRAN compilers, we have found that all of them accept quotes for hollerith output statements, even though the convention may not be in the compiler manual. The 'H' specification **will** always work, since that is ANSI standard. The use of quotes may not always work, because they are non-standard. They do, however, save the chore of counting the number of characters. If your FORTRAN compiler does not accept the use of quotes, please forgive me. If you aren't sure, EXPERIMENT! (Note that the CDC 6600 FORTRAN compiler also accepts * as a hollerith delimeter.)

Also, when we talk about our Startrek, it must be noted that the Startrek version we are talking about is ex-UCLA & CCN version that was rewritten into FORTRAN at UCLA by Vic Tolomei. Prior to UCLA, the program visited USC, ISI, and TENEX. The earliest reference in the coding refers to YALE. (BASIC SUBROUTINES FOR YALE PDP-10). We want to give our thanks to Vic Tolomei, who must have sent many hours re-writing this monster Startrek for use on UCLA-CCN TSO systems November 73 to February 74. Another name that shows up in our listings is Don D. Worth (CCN). We have a feeling (knowing?) that this version of Startrek has passed through many hands (and ALL of them anonymous). We would like to give anyone who had anything to do with this Startrek the recognition that they so justly deserve.

OF SHIPS AND SHOES AND REAL-TIME LOOPS . . .

Let's forget about the first two subjects and talk about real-time loops. (Unless the ship is the Enterprise and the shoes are Capt. Kirk's . . .)

The basic real-time loop in Startrek is shown in fig. 1. Note that all processing peculiar to a command being processed is **not** within the loop. The main concerns of the loop are ship, enemy, and torpedo movements, and any enemy action that takes place while you are away from the TVT. (Yes, if you go to answer the phone, you may be destroyed).

OF ENERGY AND KINGS . . .

Kings have always desired POWER. Question: What is POWER? Answer: ENERGY. (And if you think of cabbages as energy-storage devices, you must be a vegetarian). Anyhoo, the purpose of storing energy aboard the Enterprise is to make the whole game more realistic. The energy flow of the Enterprise is shown in Figure 2.

ACQUIRING CABBAGES [whoops, energy]:

The Enterprise can gain energy in two ways: 1) docking with a base, and 2) docking with a ghostship. In case 1), the ship is always loaded up to its maximum 5000 units of energy. In case 2), however, the energy gained is a random amount between 0 and 2000 units. If the available energy plus the energy gained is more than 5000 units, the excess energy is **not** put in the Enterprise's banks.

ENERGY STORAGE:

This section will not concern itself with a discussion of the ship's energy banks (other than as a repository of power). What we **will** talk about, however are those fantastic SHIELDS that are the protection of our fighting vessel.

In a manner known only to Com. Scott, the Energy that is put into the shields from the energy banks is DOUBLED! In other words, if you put 2000 units of energy into the shields, they will be able to soak up 4000 units of enemy torpedo and phaser hits. (When you lower your shields, you do **not** get back twice the power. That's CHEATING!)

There are, in our Startrek, (and should be in yours) penalties for flitting about without your shields up. First, the Energy loss is equal to the amount of energy in the enemy weapon (350 units for a torpedo). Second, loss (or lowering) of shields allows men to be killed and/or facilities to be damaged.

ENERGY USES: COLLISIONS:

Collisions are to be avoided like the plague. Every erg of energy in your shield is destroyed, and there are penalties for sitting around with your shields down (see above). If you collide with an object without your shields up, the penalties are the same as above, but the greater energy contained in enemy ships, bases, or ghost-ships must be absorbed. If you collide with a star, the galaxy will mourn your loss.

ENEMY ACTION:

Enemy action always takes power from the shields first. If the barrage contains more energy than the shields, the shields are destroyed and ex-

(Continued on Page 24)

Join SCCS Now!!

The Southern California Computer Society is a non-profit association whose purpose is to promote and encourage a community interest in the computer arts and sciences. Of particular interest is the application of the computer to education and to the social, scientific and environmental problems of society.

Formed in June 1975 by a handful of computer hobbyists, the Society has grown rapidly to include close to 250 consumers, amateurs and professionals. The Society serves as a focal point through which computer buffs can meet and share common interests and needs.

WHAT CAN SCCS DO FOR YOU?

- Monthly meetings for the exchange of information and computer equipment.
- A magazine (SCCS *INTERFACE*) communicating topics of interest to all members.
- Classes on programming, building computers and the general use of computers.
- Informal technical assistance and advice on anything related to computers.
- Consumer clout through group purchases.

WHO SHOULD JOIN SCCS?

- Anyone (male or female) who is interested in computers as a hobby, or
- Anyone who wants to be introduced to the world of computers (but does not know where to start), or
- Anyone who has computer related talents to share.

> "Welcome to the computer world, the damndest and craziest thing that has ever happened. But we, the computer people, are not crazy. It is you who are crazy to let us have all this fun and power to ourselves."
>
> T. H. Nelson
> *Computer Lib*

NEXT MEETING: SATURDAY
DECEMBER 13, 1975

BUILDING "S" CAFETERIA
TRW SYSTEMS
REDONDO BEACH

DOORS OPEN AT 10:00 AM
COFFEE HOUR 12:00-1:00
MEETING 1:00-4:00

BRING YOUR EQUIPMENT TO SHOW OFF, SELL OR SWAP.

MEMBERSHIP FORM

SUTHERN CALIFORNIA COMPUTER SOCIETY
P. O. BOX 987
SOUTH PASADENA, CA 91030
(213) 682-3108

NAME ______________________________

ADDRESS ______________________________

CITY ______________________________

STATE ______________________ ZIP ____________

HOME PHONE (________) ____________-____________

BUSINESS PHONE (________) ____________-____________

DUES: $10.00/YEAR **PLEASE PRINT PLAINLY**

MAKE CHECK PAYABLE TO: SOUTHERN CALIFORNIA COMPUTER SOCIETY

(Continued from Page 13)

table given is of no value and a new one for the system the reader likes to use must be constructed).

Examples:

BINARY DECIMAL

$10010_2 = (1\times2^4) + (0\times2^3) + (0\times2^2) + (0\times2^1) + (0\times2^0)$

$= 16 + 0 + 0 + 2 + 0 = 18$

$110110_2 = 2^5 + 2^4 + 0 + 2^2 + 0 = 32 + 16 + 4 + 2 = 54$

$1001_2 = 2^3 + 0 + 0 + 2^0 = 8 + 1 = 9$

Going from binary to decimal is only traveling one way on a two way street, however, since it is indeed necessary to be capable of converting decimal to binary. There are formulas which make this simple, but they are not directly applicable to systems that have more symbols than decimals. By reversing the logic of the binary to decimal conversion, we get a crude, but direct and very convenient method which requires only some subtraction, bookkeeping, and a position table. The method is this:

1. Find the largest number in the position table which is smaller than the number to be converted. This becomes the most significant (left most) position in the binary representation of the number.
2. Subtract the value of that position from the decimal representation of the number, and place a "one" in that position of the binary representation. If there is no remainder of the decimal representation of the number, then the conversion is finished.
3. If the value of the next position to the right in the binary representation of the number is greater than the remainder of the decimal number, it is too big to use, so put a "zero" in that position and repeat this step. If the value of this position is less than the decimal remainder, return to Step #2.

The method is not very sophisticated, but it requires very simple arithmetic and gives a good feeling for the number systems with which it is used.

Example: $4095_{10} = ?_2$

$2^{11} = 2048$ so position eleven becomes the most significant binary position.

	Binary No.	**Position**
4095		
−2048	1	11
2047		
−1024	1	10
1023		
−512	1	9
511		
−256	1	8
255		
−128	1	7
127		
−64	1	6
63		
−32	1	5
31		
−16	1	4
15		
−8	1	3
7		
−4	1	2
3		
−2	1	1
1		
−1	1	0
0		

$4095_{10} = 111111111111_2$

$351_{10} = ?_2$

$2^8 = 256$ so position eight becomes the most significant position.

	Binary No.	**Position**
351		
−256	1	8
95		
−0	0	7
95		
−64	1	6
31		
−0	0	5
31		
−16	1	4
15		
−8	1	3
7		
−4	1	2
3		
−2	1	1
1		
−1	1	0
0		

So, subtracting 256 from 351 = 95. The binary value of the next position (position 7) is 128 which is greater than our remainder of 95, thus a zero for position 7. Value for position 6 is 64, so 95 minus 64 equals 31, and we now have 101xxxxxx. The process is continued until the remainder is zero.

$351_{10} = 101011111_2$

$40 = ?_2$

$2^5 = 32$ so position five becomes the most significant.

	Binary No.	**Position**
40		
−32	1	5
8		
−0	0	4
8		
−8	1	3
0	0	2
	0	1
	0	0

$40_{10} = 101000_2$

The methods are identical for converting decimal to octal or hex and back again.

POSITION VALUE FOR HEX

POSITION	3	2	1	0
HEX	DECIMAL			
0	0	0	0	0
1	4,096	256	16	1
2	8,192	512	32	2
3	12,288	768	48	3
4	16,384	1,024	64	4
5	20,480	1,280	80	5
6	24,576	1,536	96	6
7	28,672	1,792	112	7
8	32,768	2,048	128	8
9	36,864	2,304	144	9
A	40,960	2,560	160	10
B	45,056	2,816	176	11
C	49,152	3,072	192	12
D	53,248	3,328	208	13
E	57,344	3,584	224	14
F	61,440	3,840	240	15

POSITION VALUE TABLE
FOR OCTAL

POSITION	5	4	3	2	1	0
OCTAL	DECIMAL					
0	0	0	0	0	0	0
1	32,768	4,096	512	64	8	1
2	65,536	8,192	1,024	128	16	2
3	98,304	12,288	1,536	192	24	3
4	131,072	16,384	2,048	256	32	4
5	163,840	20,480	2,560	320	40	5
6	196,608	24,576	3,072	384	48	6
7	229,376	28,672	3,584	448	56	7

Example:

Hex **Decimal**

$ABC4_{16}$ $= (A_{16} \times 16^3) + (B_{16} \times 16^2) + (C_{16} \times 16^2) + (4_{16} \times 16^0)$
$= (10 \times 16^3) + (11 \times 16^2) + (12 \times 16) + (4 \times 16^0)$
$= (10 \times 4096) + (11 \times 256) + (12 \times 16) + (4 \times 1)$
$= 40960 + 2816 + 192 + 4 = 43968$

Example:

$5022_{10} = ?_{16}$
$16^3 = 4096$ so 3 is the most significant position

It is possible to convert from hex to octal using this method and a new set of position value tables. The fact is though, that there exists an extremely simple set of relations between binary, octal, and hex which makes conversion from one system to the other simplicity itself. A glance at the "4 System Number Translation Table" should verify the fact that any single octal symbol can be expressed by a unique three-digit binary number, just as any single hex symbol can be represented by a unique four-digit binary number.

This shouldn't be an astounding revelation since the main premise of the article is that a number can be expressed in any number system. The important fact to note is that there are *only eight* combinations of three binary digits and *only sixteen* combinations of four binary digits. (Bells should start ringing and lights flashing because nobody said that a symbol couldn't consist of more than one figure.) There are sound reasons for this pair of coincidences: $8 = 2^3$ and $16 = 2^4$. The upshot of this is that 4 binary positions may be replaced by one hex position and 3 binary positions may be replaced by one octal position.

Example: $XXXXYYYYZZZZ_2$,
RSQ_{16}

$ZZZZ_2$ is a number greater than, or equal to 0, but less than or equal to 15. Q_{16} is a number greater than or equal to 16, but less than or equal to 240, as is S, the second digit of the hexadecimal number. XXXX is a number greater than or equal to 256, but less than or equal to 3840, as is R, the third digit of the hex number.

Octal can be treated similarly, but a few concrete conversions should more completely make the point. The "positional" method of converting the numbers still applies to what is about to be shown, but abstracting from a single character sym-

(Continued)

	Hex. No.	Position
5022		
$-4096 = (1\text{x}16^3)$	1	3
926		
$-768 = (3\text{x}16^2)$	3	2
158		
$-144 = (9\text{x}16^1)$	9	1
14		
$-14 = (14\text{x}16^0)$	E	0
0		

$5022_{10} = 139E_{16}$

Octal Decimal

$567_8 = (5\text{x}8^2) + (6\text{x}8^1) + (7\text{x}8^0)$
$= (5\text{x}64) + (6\text{x}8) + (7\text{x}1)$

$= 329_{10}$

$5022_{10} = ?_8$
$8^4 = 4096$ so 4 is the most significant position.

	Octal No.	Position
5022		
$-4096 = (1\text{x}8^4)$	1	4
926		
$-512 = (1\text{x}8^3)$	1	3
414		
$-384 = (6\text{x}8^2)$	6	2
30		
$-24 = (3\text{x}8^1)$	3	1
6		
$-6 = (6\text{x}8^0)$	6	0
0		

$5022_{10} = 11636_8$

bol to a multiple character symbol is hard to do without interaction not provided by an article. So study the example with a mind to learning the trick and hopefully by the time it has been mastered, its connection with the rest of the article will become apparent.

Examples:

$5022_{10} = 11636_8 = ?_2$

1	1	6	3	6
001	001	110	011	110_2

Binary Decimal

$001001110011110^2 = (1\text{x}2^{12}) + (1\text{x}2^9) + (1\text{x}2^8) + (1\text{x}2^7) + (1\text{x}2^4)$
$+ (1\text{x}2^{12}) + (1\text{x}2^9) + (1\text{x}2^8) + (1\text{x}2^7) + (1\text{x}2^4)$
$= 4096 + 512 + 256 + 128 + 16 + 8 + 4 + 2$
$+ 5022_{10}$

$5022_{10} = 139E_{16} = ?_2$

1	3	9	E
0001	0011	1001	1110_2

$0001001110011110_2 = 5022_{10}$
(proof above)

Converting from binary to octal or hex is even easier. To convert to octal, start from the rightmost digit and convert every three digits to their octal equivalent (filling out to the left with zeros is permissible since they don't change the value). To convert hex, start from the rightmost digit and convert every four digits to their hex equivalent.

Examples:

0001	0011	1001	1110	001	001	110	011	110
1	3	9	E	1	1	6	3	6

$139E_{16} = 0001001110011110_2 = 011636_8$

1101	0001	0000	1100	1	101	000	100	001	100
D	1	0	C	1	5	0	4	1	4

$D10C_{16} = 1101000100001100_2 = 150414_{16}$

Finally, we supply a few cases which can be used as exercises with answers to sharpen technique.

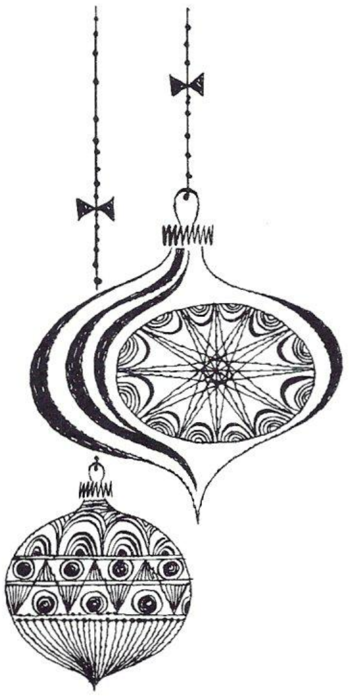
$27045_{10} = 69A5_{16} = 064645_8$
$= 0110100110100101_2$

$9411_{10} = 022303_8 = 24C3_{16}$
$= 0010010011000011_2$

$42747_{10} = 1010011011111011_2$
$= 123373_8 = A6FB_{16}$

$4660_{10} = 011064_8$ —
$0001001000110100_2 = 1234_{16}$

The reader should now have an understanding of the basics of number systems as applied to digital computers. The practice obtained in the everyday use of assembly listings on either 8 bit or 16 bit machines will soon leave the reader proficient in conversion of decimal to octal or hexadecimal representations and vice versa.



(Continued from Page 10)

This subroutine call would transfer a block of data between a buffer space in memory and a particular location on the assigned device. The DEVICEPOINTER value would refer to a table in the system which has been set up with the necessary information to refer to the correct device handler.

In the case of the standard, non-buffered call as described in solution 3. above, it is only necessary to communicate the device pointer (referring to one of a small set of devices accomodated by the system), the READ/WRITE bit, and probably leave the accumulator free to transmit the data byte.

```
JMPSUBR    CHARIO
R/W    DEVICEPOINTER
program code continues

(AC contains 1 byte of data)
```

What else should the system do?

If the monitor deals with operator interaction, the rest of the system deals with demands from programs running under the system. In some cases, both of these "users" of the system want the same things and in other cases their demands on the system do not overlap. Things which programs will need to do, but the operator will not usually need directly from keyboard interaction are file blocking and unblocking, various kinds of format conversions such as binary-to-decimal and vice-versa, and other frequently needed operations which the manufacturer left out of the chip's instruction set. Routines to do these tasks can usually be ordinary subroutines coded in the resident system area (always in memory). Routines to perform tasks needed both by pograms and by the operator may be similar in form, but may have to have more general calling sequences. The best examples of operations needed both internally and externally are those concerned with file directory maintenance. Returning to the problem of deleting a file, we can now see that the cumbersome task of deciphering and modifying the actual directory structure should be coded only once and should be accessible as a system routine. This routine can then be called by other parts of the system or by separate programs as needed. Let's now take a brief look at this file manipulation task in somewhat more detail.

File Directory operations

As soon as you have a device capable of holding more than one program or string of data, you have to do something to keep it all straight. What you do is reflected in the system's operation from sign-on till shut-down from then on, and must be considered carefully. There is a lot to say about this and I hope to continue the discussion in another article. For now, I will just outline the basic tasks and suggest some of the choices which could be made.

Look-up: you want to see if a file with a certain name exists on a certain device. If it is there, this routine returns the starting block of the file on that device, the length of the file in blocks and possibly other information such as the type of file it is, the date and time it was created, etc.

Create: you want to open a space on a device to write a new file. You may or may not know at that time just how much space you will finally need. This routine puts the new name in the directory, maybe in a temporary form (an open file), and tells you how much space is available and possible whether that space is all in one chunk or scattered out among various blocks on the device (linked files are an extra complication).

Close: a temporary open file is closed with a certain length and the temporary name is made permanent. That is to say that pointers to the currently active spot in the file may be discarded. The directory must be updated to reflect any empty blocks left in the device and the date and time may be entered with the new name.

Delete: Files could have attributes of deleteability depending on various conditions which must be checked. If it's OK to delete it, the directory must reflect the new empty block situation.

Passing file parameters

Hand in hand with general device I/O and file directory capability comes the problem of communicating device and file assignments to running programs. A section of the resident system tables is dedicated to this job. This file parameter table may be set up by any program or by the system, in particular, under control of the monitor command analyzer. The file table will be read by any program which calls for standardized system I/O. The actual location of and communication with the file could be done within the system or by the program itself, calling system directory and I/O routines. The file table must allow for both input and output files which must be handled differently but may not need to be segregated in the table.

In addition to device and file names, it is often convenient if the table includes space for some extra parameter information, a few numerical arguments, possibly a set of binary flags and perhaps a bit of text. These arguments can be used in different ways specific to each program and come in handy for intercommunication between programs as well as when starting a program by monitor command.

Multi-tasking ??

I will not take further time now to discuss the possibilities of setting up a multi-task environment in a microcomputer. There are some who will want this for sure, and it would be nice to have a modem on line, for example, open for incoming calls without bothering you, deep in concen-

(Continued)

(Continued from Page 23)

tration at your space-war game. I remain open to further discussion on this question. As far as designing an all-around general use operating system, my feeling is that this is one area where the appropriate solution is in having two versions, one multitask and one not.

* * *

(Continued from Page 18)

cess energy is taken from the energy banks on a one-for-one basis. In addition, men and/or facilities get damaged.

MOVEMENT:

In the following equations, we will follow these conventions:

E — current energy
E' — energy after movement
S — speed of the Enterprise
C — change of speed
A — angular change in radians

For a change in speed, the following formula is used:

$$E' = E - (20C)^2$$

For a change in direction, the following formula is used:

$$E' = E - (1 + 5AS)^2$$

The two formulas given are for normal maneuvering. There are two other types of movement that require energy: Evasive maneuvers and Emergency maneuvers. These take 200 and 250 units of power respectively.

All of the energy used in movement is obtained from the banks. What happens if we desire to move and there is not enough energy in the banks to accomodate our wishes? A good question that must be answered. Remember fig. 2? AHA! We drop our shields and get back all of that stuff we need to move. It's good to remember that whenever the level of energy in the banks drops to zero, the shields will be dropped automatically.

WEAPONS:

Torpedos take one unit of energy to launch. Phasers will take as much energy from the banks as we request. If we ask for more energy than is in the banks, the request is not honored, and the shields are not dropped.

LIFE SUPPORT:

Life support in the Enterprise uses a minimal amount of energy. In our version of Startrek, it amounts to one unit of energy per pass through the real-time loop.

figure 2.

ENERGY FLOW IN THE ENTERPRISE

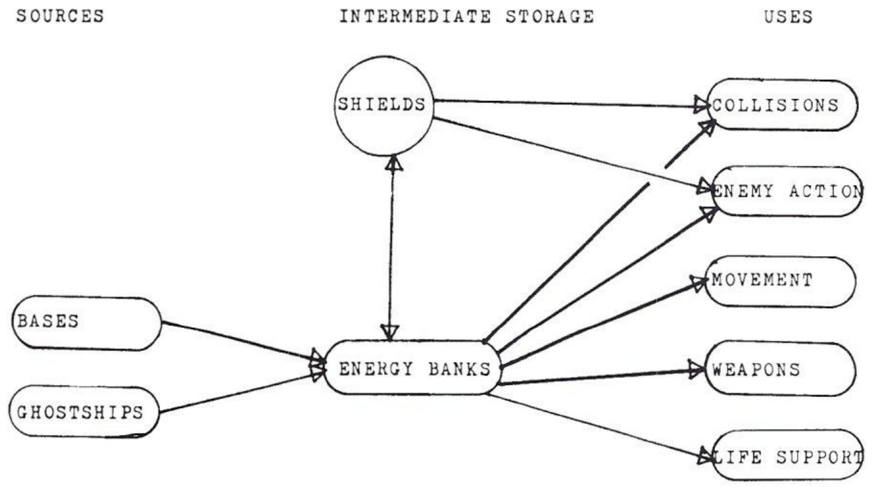
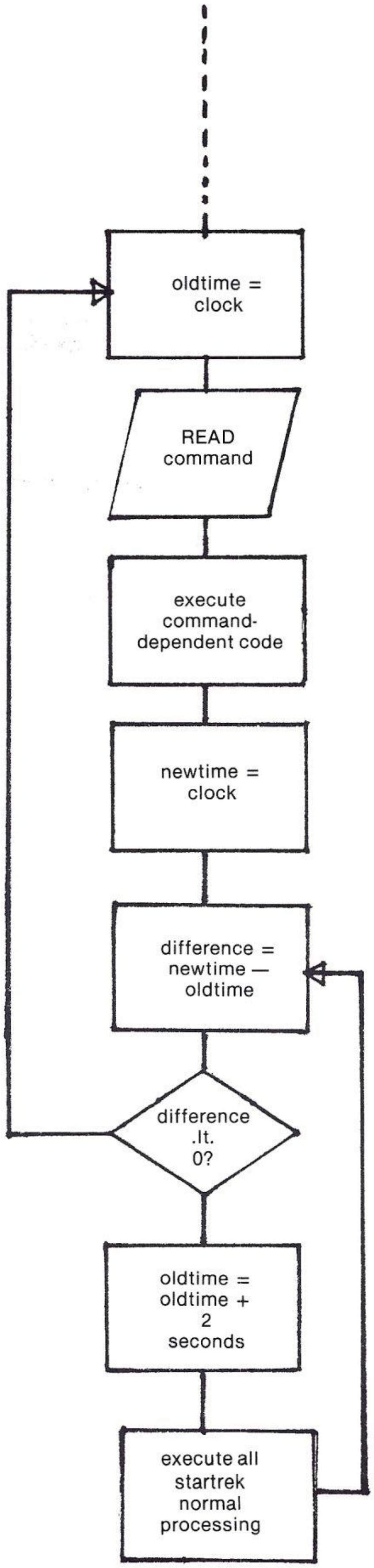

SOURCES
INTERMEDIATE STORAGE
USES
SHIELDS
COLLISIONS
ENEMY ACTION
MOVEMENT
BASES
ENERGY BANKS
WEAPONS
GHOSTSHIPS
LIFE SUPPORT

Figure 1.
THE REAL-TIME LOOP IN STARTREK

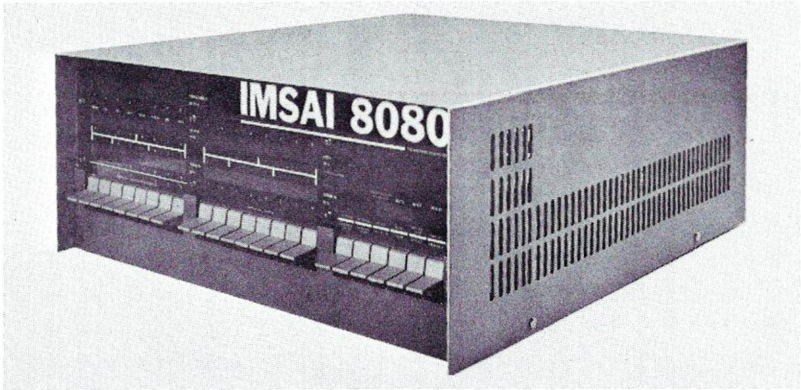

oldtime = clock
READ command
execute command-dependent code
newtime = clock
difference = newtime — oldtime
difference .lt. 0?
oldtime = oldtime + 2 seconds
execute all startrek normal processing

COMING SOON

THE COMPUTER MART

One — stop computer supermarket. —Featuring —

THE IMSAI 8080 The super computer.



The Altair is a good idea — but —

THE IMSAI 8080 is even better! ! ! ! ! !

WHEN YOU BUY THE IMSAI 8080 YOU GET ALL OF THE GOOD FEATURES OF THE ALTAIR PLUS.

1. A commercial grade cabinet. Rack mounting available.
2. Smoked lucite display panel with non-marring legends.
3. Rugged commercial grade paddle switches with reliable debounce circuits.
4. Gold plated contacts on board edges.
5. Heavy duty power supply to handle extra cards. 24 amperes at 8 volts, 3 amp each at + 16 and — 16 volts.
6. Sturdy card cage to properly support all circuit boards.
7. Plus removeable front panel that eliminates normal wiring harness between front panel and back plane.
8. Full line of peripheral devices to build truly powerful and complete systems.
9. An exclusive shared memory facility to allow multiple microprocessors in the same cabinet and share the same memory.

All IMSAI 8080 and Altair 8800 boards and software are plug-in compatible, and interchangeable without modification.

BEFORE YOU BUY AN ALTAIR, YOU OWE IT TO YOURSELF TO SEE THE IMSAI 8080! The IMSAI 8080 costs no more, yet gives you much more.

Buy the BEST, Buy an IMSAI 8080

THE COMPUTER MART

Authorized Dealer

Suite 307, 2333 Beverly Boulevard • Los Angeles 90057 • (213) 484-2002

Interfacial

INTERFACIAL — The editor's aims, observations and opinions

Most, if not all, members of the Southern California Computer Society have built, are building, or are planning to build their own computer, and upon completion, plan to program it to perform various functions designed to astound, confound or just plan delight.

The staff of SCCS INTERFACE is composed of members who joined the club because of the desire to do just that: play computer freak, not play journalist. Several steps are being taken to facilitate the staff's return to computers; i.e., expansion of the editorial staff to reduce the time required of any one member, invitation for contributions from anyone with something of interest to say to the computer experimenter or hobbyist, and last, but not least, a contractual arrangement with a firm to handle the mechanical (typesetting, paste-up of publication, printing, binding, distribution, etc.) The latter of these moves requires money, hence the advertisements.

The Board of Directors gave serious consideration to the solicitation of advertising, as they were aware that some members would be opposed to making a "commercial publication" of the club newsletter. The result of much heated debate and several meetings devoted primarily to the matter was the concensus that the benefits to readers of SCCS INTERFACE will be well worth the risk of any adverse reaction. Further, well done advertisements provide a good mechanism for keeping abreast of the results of competition (visions of a 25 cent 2102?) and the latest in technical advances.

So — here we go — starting out with a series of articles on that most important machine, the teletype, which will include how it came about, how it works **and** how to maintain and repair it, all written by Cliff Sparks. Cliff's many years' experience with the device, as well as his offer to do the series, will prove beneficial to any hobbyist who owns one.

Jeff Landry's article on number systems, and Lloyd Rice's article on the design of a micro-operating system are the software contributions for this issue. In the hardware department is a reply to a plea for help from a software type by Mike Thompson. His cursor modification for a TV typewriter were demonstrated to this writer and Mike kindly consented to share it.

Several interesting articles for future issues are in the works, including a series on micro-processor theory and design for the beginner. The staff is querying members on their projects, hoping to find lots of interesting material to publish. Any and all readers are invited to contribute, as well as respond to that which has been published. All levels of contribution are welcome: from beginners' experiences and perceptions to expert designer's creations, from hardware to software, from articles which help the novice educate himself to articles on the latest in state of the art.

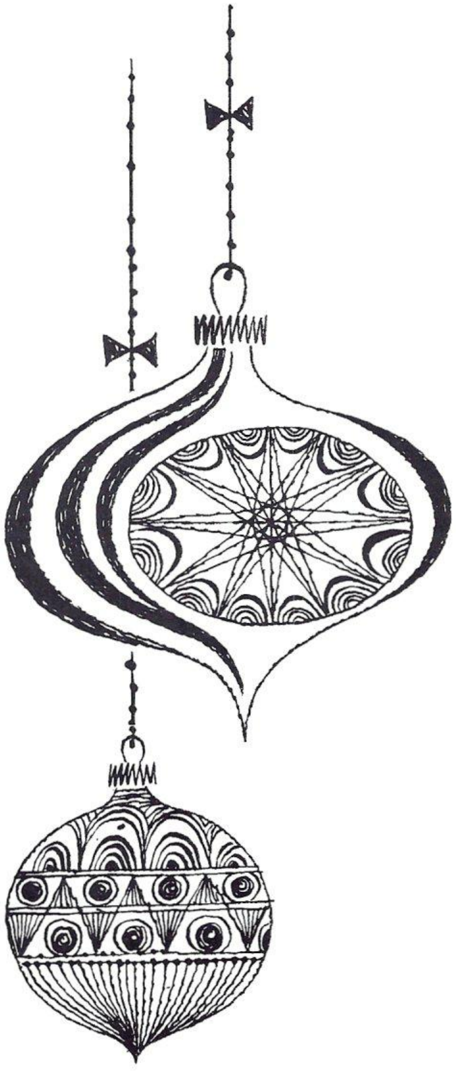
Members are invited to utilize the free ad privilege — non-members are invited to join SCCS. All are invited to let the staff know what your desires and reactions are. Letters to INTERFACE will be published as space permits.

One final item in a more personal vein before I return to my half-finished Altair. It's the aim of the SCCS INTERFACE staff to at least try to be all things to all computer experimentors and hobbyists. We want to help the seeker of advice and assistance find the giver, help the novice get started, whether he needs help with hardware, software or simply direction, and help the more experienced computer nut find fellow enthusiasts of equal experience to compare notes, programs and hardware ideas.

We don't aspire to be another BYTE, but we do want to reach as far and wide as the U.S. Postal Service will allow. It makes no difference if you grow potatoes in Idaho, manufacture cars in Detroit, chase cows and oil wells in Texas, or simply enjoy retirement in Florida. You are invited to become a member of SCCS, a reader of INTERFACE and a member of that rapidly growing community who are discovering IBM has no monopoly on computers!

We're looking forward to hearing from you. Happy Computing.

Art Childs, *Editor*



Letters to the Editor

MITS—WATCHING IN COUNCIL BLUFFS

First off, I'm a beginner trying to get some workable programs for the 1K ALTAIR 8800 I built from the kit. I also have a beautiful Processor Technology 3P+S I/O interface kit in the box, just waiting for the solder iron. Since I had trouble with a master-slave flip-flop on the 1K Memory Board, I think I had better use Molex pins for all the IC's on the interface board. Is there a good reason not to use Molex pins?

Second, thanks for the advance copies of INTERFACE. I'll be watching for the non-member subscription information, because INTERFACE is the best of all I receive — Peoples Computer Company, The Computer Hobbyist, Mits Computer Notes, Popular Electronics, Radio Electronics and Byte.

Enjoyed reading Tom Gallant's "Joys (?) of Building an Altair." The MITS sand colored sheet labelled OCTAL PROGRAMMING SET 8800 contains two errors. The MVI r OD6 and PCHL 351. In the MITS Operators Manual page 3, paragraph 3, it is obvious that their typist left out an entire line of the rough draft. It probably should read, "The AND statement is true, if and only if, all of it's logic statements are true. The OR statement is true if either or all of its logic statements are true."

Keep up the fine work!

Walt Gerard

EDITORS NOTE:

We called MITS to ask them about the items in Walt's letter. They were in agreement about the errors in the instructions set, but disagreed on the matter of the logic, laying the cause of the trouble to poor sentence structure.

Nothing was said about an errata sheet to cover the errors on the wording of the sentence on logic, but they did drop hints of good things coming in the near future for purchasers of MITS kits and equipment—not the least of which is an all-out effort to improve delivery time. It sounds like they intend to be the fastest as well as the biggest.

We queried several hardware types and the response was: use a regular I.C. socket—there are many good quality gold plated sockets manufactured.

The Board of Directors has made no provision for non-member subscriptions, Walt, but your sharp eye [and $10] qualifies you. We will be happy to welcome you as the first Council Bluffs member to SCCS.

SMOKE SIGNALS ANSWERED

EDITORS NOTE:

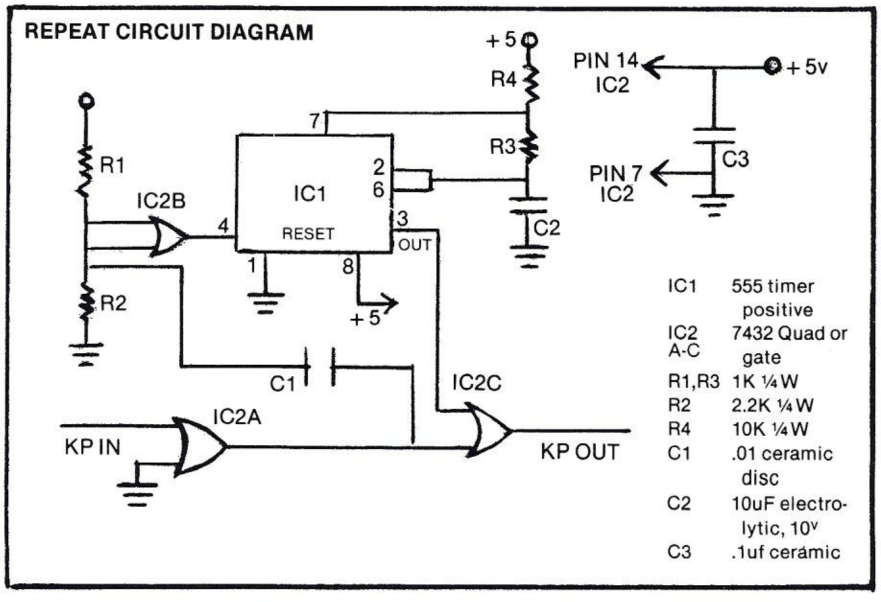
Mike Thompson answered Art Childs' plea for help [see "Binary Smoke Signals", October Newsletter Issue] with the circuit described below. It works, and it appears simple enough so that even a software-type computer freak should be able to build it.

This simple circuit adds repeat action to your keyboard. When a key is held down, the keypressed line is pulsed causing the character to repeat until the key is released. Due to its rather simple nature, if you don't release your keys quickly, you may get repeats where you don't really want them.

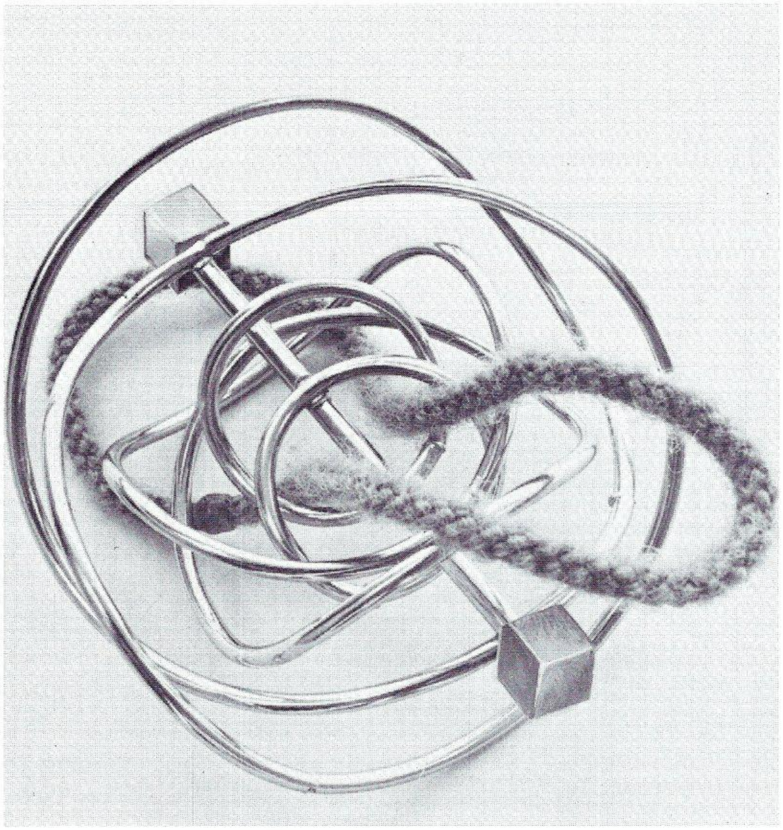
IC1 is hooked up as an astable multivibrator with a free running frequency around 8 Hz. The KP signal is buffered by IC2A. IC2C is a negative logic AND gate which "ands" the KP signal with the output of IC1. When the normally high KP strobe goes low, a pulse is delivered through C1, buffered by IC2B, to the reset input of IC1. This sets the output of IC1 low, causing the KP output to drop. The pulse from IC2B is brief, so IC1 will start to oscillate again. The KP out signal will follow IC1's output as long as the KP in signal is low. The frequency of IC1 can be varied by changing R4 of C2.

This circuit was tested with a SWTP KBD-3 keyboard and should work with any keyboard whose keypressed line is TTL compatible, normally high, which goes low and stays low, and when a key is pressed. If the initial repeat action occurs too quickly, causing double characters when only one is desired, try adjusting the value of C1.

REPEAT CIRCUIT DIAGRAM



IC1	555 timer positive
IC2 A-C	7432 Quad or gate
R1,R3	1K ¼W
R2	2.2K ¼W
R4	10K ¼W
C1	.01 ceramic disc
C2	10uF electrolytic, 10^{v}
C3	.1uf ceramic



"THE WIZZIT"

Designed for the discerning, the ultimate challenge, the unmerciful mind boggler.

Rugged construction to withstand most instantaneous aerial launches toward solid objects.

This puzzle is a must for your game collection. The object is to remove the loop from its wire web. (No cutting or bending please.) It really can be removed —if you give up easily we can supply the solution. Bet you can't do it! Try it!

Only $5.95 and it's all yours! ORDER ONE NOW!!! Send check or money order to: "THE WIZZIT" P.O. Box 1234, Cerritos, CA 90701

California Residents Add 6% Sales Tax.

Classified Advertising

RGS ELECTRONICS

Microconputer Kits
Surplus ICs, Capacitors,
Resistors, etc.

2K RAM Board Kit
Includs 2K 8 2102s
Bypass Capacitors
and Edge Connector
$54.00 After Discount

Cal.residents add sales tax
Send stamp for free flyer
3650 Charles St., Suite K
Santa Clara, CA
95050
(408) 247-1058

MICROCOMPUTER PRODUCTS

- GP MICROCOMPUTERS
- CUSTOM CONTROLLERS
- PROM/ROM PROGRAMMERS
- MICROCOMPUTER MODULES
- MICROCOMPUTER KITS
- 8080A, 1702A, 2102-X IC's
- SOFTWARE ASSEMBLERS
- UTILITY SYSTEMS

MICROCOMPUTER SERVICES

- CONCEPT DEVELOPMENT
- REQUIREMENTS ANALYSIS
- HARDWARE DEVELOPMENT
- PRODUCT DEVELOPMENT
- APPLICATION PROGRAMMING
- LAB SPACE RENTALS
- PROM PROGRAMMING
- MICROCOMPUTER CONSULTING

AUTOMATED COMPUTER SYSTEMS

2361 E. FOOTHILL BLVD. • PASADENA, CA. 91107 • TELEPHONE (213) 792-1193

McPheters, Wolfe & Jones MWJ

ADVERTISING MARKETING PUBLIC RELATIONS
6515 SUNSET BLVD. SUITE 202
469-7789
HOLLYWOOD, CALIF. 90028

LIBE BULLETIN #2 NOW AVAILABLE. LEARN HOW TO SOLVE HIGHER-ORDER EQUATIONS ON A HAND CALCULATOR. BASIC EXPLANATIONS, NON-PROGRAMMABLE OR PROGRAMMABLE CALCULATORS. SEND $1.25 (CALIF. RESIDENTS INCLUDE 6% STATE TAX) TO: LIBE COMPANY, P.O. BOX 1196, LOS ALTOS, CA 94022.

MEMORIES ARE MADE OF THESE:

INTEGRATED CIRCUITS

2102 1K STATIC RAM---STILL ONLY $1.95 GUARANTEED FASTER THAN 750 NS. BUY 100, TAKE 20%; BUY 1000, TAKE 30%.

5203 2K EROM---$9.50. THIS STATIC MEMORY IS FULLY PROGRAMMABLE AND MAY BE ERASED WITH UV LIGHT. NON-VOLATILE.

5204 4K EROM---$24.50. SIMILAR TO THE 5203, BUT 4K OF MEMORY.

MEMORY/PROCESSOR 8 BIT CHIP SET: $32.50 BRINGS YOU 8 OF OUR 2102S AND AN 8008 MICROPROCESSOR.

EROM BOARD KITS

8K BY 8 BOARD...$352 (ADD $48 & WE PROGRAM)
4K BY 8 BOARD...$200 (ADD $25 & WE PROGRAM)
2K BY 8 BOARD...$125 (ADD $15 & WE PROGRAM)

Now you can stuff up to 8K of your favorite software----editor/assembler or what have you----on a board. Program it yourself or have us do the programming. Just like the RAM kit, DIRECT ALTAIR 8800 PLUG-IN COMPATIBLE. Includes sockets, bypassing, industrial-quality plated-through board, buffered addresses and outputs, on board regulation. LOW POWER: Full 8K requires ½A @ 5V, 150 ma @ -12V. File protect feature includes line receiver with real hysteresis to prevent false triggering on noise. EXPANDABLE: If you buy our 2K board and want to move on to bigger and better things (like 8K), all you add is more sockets and EROMS. Your board doesn't become obsolete.

4K BY 8 RAM BOARD $109.22

ONE-THIRD OF A CENT PER BIT---Our most popular memory kit. You get sockets for all ICs, industrial-quality plated-through board, lots of bypassing, 5 voltage regulators to share the power load (less thermal problems, more reliability), typical 500 ns access time at 25°C, and buffered addresses and outputs (no input presents more than 1 LPTTL load; outputs can drive 20 standard TTL loads). May be powered with 8 volts unregulated; or by bypassing the on board regulators, with +5V. Complete with assembly instructions and logic print of the RAM board.

ALTAIR 8800 OWNERS, PLEASE NOTE: THIS MEMORY BOARD KIT, LIKE OUR OTHER MEMORY KITS, IS DIRECTLY PLUG-IN COMPATIBLE WITH THE 8800.

EROM BOARD SPECIAL OFFER

We've got 4K by 8 boards pre-programmed with 8080 assembler, editor, and monitor routines. $200 buys you the same quality and ALTAIR PLUG-IN COMPATIBILITY as our other memory kits. Another advantage: because this is EROM rather than mask programming, you can make changes if desired. Speaking of changes, although these routines work our really well, we are including 2 free updates to purchasers should bugs or modifications develop. Any additional changes are available for a very nominal charge. SEND $2.95 (PPD) FOR OUR SOFTWARE INFO PACKET IF YOU WANT MORE INFO OR WANT TO LOAD YOUR OWN ROM BOARD. INCLUDES INSTRUCTION LISTING, SCHEMATIC, ASSEMBLY DATA, ETC.

LOW POWER SCHOTTKY

COMPARED TO STANDARD TTL, LOW POWER SCHOTTKY OFFERS 1/5 THE POWER CONSUMPTION-BUT WITH NO SPEED PENALTY. CONSIDERED BY MANY AS THE LOGICAL SUCCESSOR TO STANDARD TTL.

74LS00	Quad 2 input NAND gate	.40
74LS04	Hex inverter	.45
74LS08	Quad 2 input AND gate	.40
74LS10	Triple 3 in NAND gate	.40
74LS20	Dual 4 input NAND gate	.40
74LS42	BCD to decimal decode	1.25
74LS138	1 of 8 decoder	1.40
74LS168	Dec Syn Up/Down cntr	1.50
74LS169	Bin Syn Up/Down cntr	1.50
74LS175	Quad latch	1.38

Hard to find TTL

74273	8 bit latch (20 pin)	1.89
74365	Noninv 3 st buf NOR en	.63
74366	Inv 3 state buf NOR en	.63
74367	Noninv 4-2 buf w/ sep en	.63
74368	Inv 4-2 buf w/ sep en	.63
74390	16 pin dual decade cnt	1.50
74393	14 pin dual binary cntr	1.50

MICROPROCESSORS

16 BIT CHIP SET $195

1 • PACE CPU
4 • DS3608
1 • DS0026
1 • DM8837
32 • 2102 (4k bytes RAM)
2 • MM5204 (1k byte ROM)
AND OUR PACE DATA PACKET!

A POWERFUL 16 BIT PROCESSOR. 40 PIN DIP; CAPABILITY FOR 45 CLASSES OF INSTRUCTIONS AND UP TO 337 INSTRUCTIONS. INTERFACE CHIPS PROVIDE TTL COMPATIBILITY--RAMS AND ROMS PROVIDE MEMORY, AND THE PACE DATA PACKET TELLS YOU HOW TO APPLY EVERYTHING.

PACE DATA PACKET AVAILABLE SEPARATELY FOR $2.50 (COVERS PRINTING AND POSTAGE). 80 PAGES OF INFORMATION ON THE CHIP ITSELF, SOFTWARE, SYSTEM ORGANIZATION, AND MORE. ALSO INCLUDES AN 11" BY 17" INCH FOLDOUT LOGIC PRINT OF PACE IMPLEMENTATION.

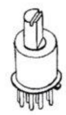
IF YOUR INTEREST IS MICROCOMPUTERS, AND PARTICULARLY PACE, YOU WILL WELCOME THIS DATA PACKET. PURCHASE PRICE REFUNDABLE WITH PACE ORDER.

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

TERMS: Add postage where indicated; add 50¢ to orders under $10. No COD orders; it's a lot of paperwork for us. Californians add tax. To place Bankamericard® or Mastercharge® orders, call (415) 357-7007, 24 hours a day. Please remember this is an answering service, and is not set up to handle technical questions.
Want to know about all the other stuff we sell? Send for our FREE FLYER!

TINY Rotary Switch



SINGLE POLE, SEVEN POSITION + OFF. IN TO-5 CAN; TINY FOR CIRCUIT BOARD MOUNTING. GOOD FOR PROGRAMMING & MUCH MORE.
$1.25 EA OR 3/$2.95

NOTE TO CLUB MEMBERS and INTERFACE READERS:

Several of you have heard about the quality of our boards; but some of you may not know that we offer discounts for quantity to individuals, groups, and clubs. POOL ORDERS AND SAVE---10% if you buy 10 boards, 20% for 100. Over 100? Call for quote.

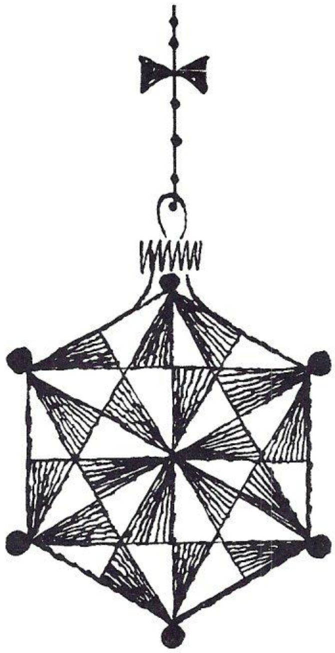
AND IT PAYS TO READ INTERFACE, TOO--5% off on any of our EROM board kits (including the 8080 software board) if you reference INTERFACE and your order is postmarked by December 31.

YOU ASKED FOR IT: We've gotten a few queries as to why we buffer the addresses and outputs of our memory boards. An unbuffered signal can do OK if you're only driving a little bit of memory---but if you want to feed lots of 2102s, you must have some extra drive in order to overcome the capacitive load of bunches of 2102s.

Unclassified Advertising.....

Caveat Emptor

POWER SUPPLY I have a quantity of 5v-6amp highly regulated power supplies taken from keyboard terminals. They also supply 12v, 48v, and 200v each at 1 amp. Full schematics supplied and plans for obtaining -9v, -12v, -5v. $25 plus postage on 15 lbs. and 6% in California. Grant Runyan, 1146 Nirvana Rd., Santa Barbara, CA 93101.



WHO NEEDS IBM? Are you using a DEC system-10? If you are interested in starting a club solely devoted to the 10, please call Ralph Klestadt at 784-8319 or 789-9616 after 4 p.m. Remember, you are not alone out there with your 10!

EXCELLENT OPPORTUNITY for over the counter sales of Microprocessors, Kits, Peripherals and software. Individual selected must be capable of guiding customers through selection of available equipment and organizing after sale training sessions for hardware and software utilization. Position requires strong background in microprocessor technology and sales experience. Send Resume and salary history to: P. O. Box 2208, Culver City, California 90230.

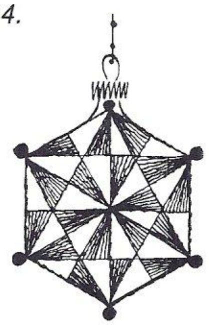
MICROCOMPUTER HELP
WANTED FOR PART OF
THE ACTION
—Part-time OK—

- *Microcomputer hardware Design Engineer*
- *Microcomputer software Development Engineer*
- *Microcomputer systems Applications Engineer*
- *Microcomputer Sales Engineer*

AUTOMATED COMPUTER
SYSTEMS
2361 E. Foothill Blvd., Pasadena,
Calif. 91107 (213) 792-1193.

NUMBER OF DAYS PROGRAM How many days til Christmas? Between 1/15/74 and 6/30/76? This little Altair program will provide the answer. I am using the sub-routine in a scheduling system to tell how many days are left to meet a schedule date. If you want a write up on the logic and a paper tape send a buck to Bill Roch, 5133 Catalon Ave., Woodland Hills, Calif. 91364.

"HANDY DANDY" PAPER TAPE WINDER $15. Bill Roch, 5133 Catalon Ave., Woodland Hills 91364 (213) 348-4278 or (213) 887-4944.



RPC-400 COMPUTER WITH PUNCH, READER and typewriter. Best offer. 1 core $15. 8 x 1K core memory with PS $35. Varian chassis with power supply $50. Several readers, printers, tape reqinds $15 more or less. Need varian DM-122 printed circuit board. Call (213) 796-8270, Bill Pfeiffer.

Each month SCCS Interface will devote this space free, non-commercial advertising by members. This is done as a service to members of the Southern California Computer Society and to help promote communications between computer experimenters and hobbyists throughout the world.

SCCS INTERFACE READER SERVICE

For your convenience, you may receive further product information by circling the appropriate advertiser index numbers and returning the form to:

INTERFACE HOTLINE
P. O. BOX 1234
CERRITOS, CALIF. 90701

NO.	ADVERTISER	PAGE
1	AUTOMATED COMPUTER SYSTEMS	Pg. 28
2	COMPUTER MART	Pg. 25
3	GODBOUT ELECTRONICS	Pg. 29
4	I. C. ELECTRONICS	Pg. 32-CIII
5	LIBE CO.	Pg. 28
6	McPHETERS, WOLFE & JONES	Pg. 28
7	MITS	Pg. 9
8	RGS ELECTRONICS	Pg. 28
9	SCCS	Pg. 19
10	SPHERE	Pg. 7
11	THE COMPUTER STORE	Pg. 16
12	THE WIZZIT	Pg. 28
13	WAVEMATE	Pg. 5

INTERFACE HOTLINE
P. O. Box 1234
Cerritos, CA 90701
Please print or type.

DEC 1975
ADVERTISER NO.
1 2 3 4 5 6 7
8 9 10 11 12 13

Name____________________

Address____________________

City__________ State__________ Zip________

new breadboard test equipment

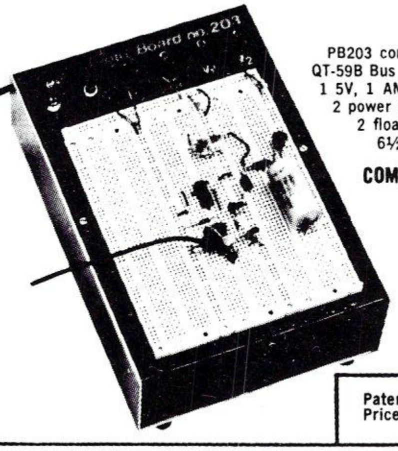
. . . with power for the professional . . . with economy for the hobbyist!

Fresh, from the laboratories of Continental Specialties, whose QT Sockets and Proto-Boards have taken the electronics market by storm'. . . here are exciting NEW ideas to make your circuit design and testing faster, safer, easier and less expensive!

NOW . . . PROTO POWER!

NEW PROTO BOARD-203

New Proto Board 203 with built-in regulated short-proof 5V, 1AMP power supply. Ready-to-use. Just plug-in and start building! 2 extra floating 5-way binding posts for external signals. Completely self-contained with power switch indicator lamp and power fuse. 24-14 pin DIP capacity. All metal construction . . . no chipping or cracking like plastic cases Two-tone gold/black quality case makes PB203 asthetically, as well as technically, pleasing.



PB203 contains: 3 QT-59S Sockets; 4 QT-59B Bus Strips; 1 QT-47B Bus Strip; 1 5V, 1 AMP regulated power supply; 2 power supply 5-way binding posts; 2 floating binding posts; 9¾"L x 6½"W x 2¾"H. Weight: 5 lbs.

COMPLETE $75

ORDER TODAY!

Patents Pending • Made in U.S.A.
Prices subject to change

For the economy-minded student or experimenter . . .

NEW PROTO BOARD-100

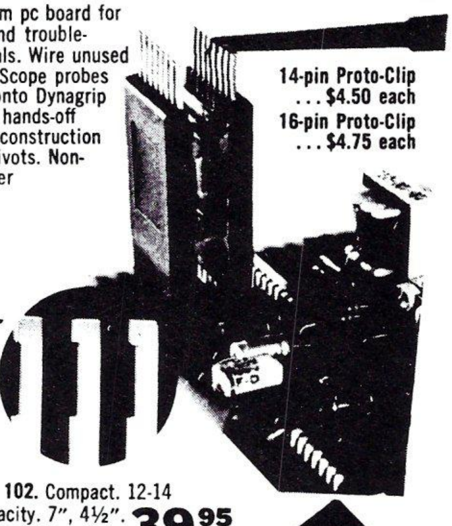
Here's a low cost, big 10 IC capacity breadboard kit with all the quality of QT Sockets and the best of the Proto-Board series . . . complete down to the last nut, bolt and screw. Includes 2 QT-35S Sockets; 1 QT-35B Bus Strip; 2 5-way binding posts; 4 rubber feet; screws, nuts, bolts; and easy assembly instructions.

COMPLETE KIT . . .

$19.95 **ORDER YOURS TODAY!**

NEW! PROTO-CLIP for power-on, hands-off signal tracing. Eliminates shorting leads . . . and costs under $5!

Bring IC leads up from pc board for fast signal tracing and trouble-shooting. Inject signals. Wire unused circuits into boards. Scope probes and test leads lock onto Dynagrip (see circle) inset for hands-off testing. New plastic construction eliminates springs, pivots. Non-corrosive nickel/silver contacts for simultaneous low resistance connections. Narrow throat for high density pc boards.

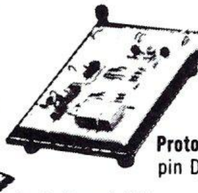


14-pin Proto-Clip . . . $4.50 each
16-pin Proto-Clip . . . $4.75 each

Build and test circuits as fast as you can think without soldering or patch cords with NEW Proto Board

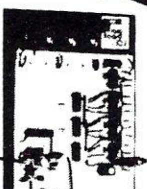
Breadboard Assemblies

Here are four versatile new Continental Specialties' Proto Boards, made from convenient combinations of QT Sockets and Bus Strips, that let you make all circuit and power interconnections with common solid #22 wire, while power distribution busses make wiring a snap. Aluminum base plates offer solid work surfaces and perfect ground plane. Rubber feet prevent scratching. Each Proto Board features one or more 5-way binding posts to tie into system or power supply ground. And all are compatible with digital and linear ICs, in TO5s, DIP packs and discrete components. Each is assembled and ready-to-use.

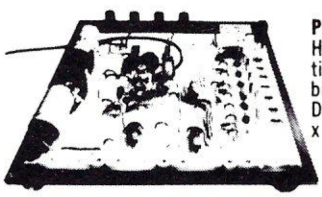


Proto Board 102. Compact. 12-14 pin DIP capacity. 7", 4½". **39.95**

Proto Board 101. Compact. Inexpensive. 10-14 pin DIP capacity. 5.8" long x 4.5" wide. **29.95**



Proto Board 103. 2,250 solderless tie points. (4) 5-way binding posts. 24-14 pin DIP capacity. 9" x 6". **59.95**



Proto Board 104. Huge. 3,060 solderless tie points. (4) 5-way binding posts. 32-14 pin DIP capacity. 9½" long x 8" wide. **79.95**

FREE!

14 PIN PROTO-CLIP
WITH ANY CSC ORDER
BY MENTIONING
INTERFACE
A $4.50 VALUE

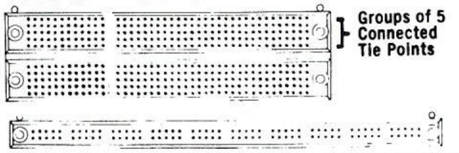
ORDER TODAY!
FROM I.C. ELECTRONIC SUPPLY
BY PHONE, MAIL OR BETTER YET
PLACE YOUR ORDER IN PERSON

OFFER GOOD THRU. JAN 76

ProtoTest Devices

Now you can breadboard circuits as fast as you can draw them!!! Plug-in . . . wire . . . test . . . modify . . . build without soldering or patch cords!!!!

Keep going non-stop, from idea stage to final design, using the same components over again without bending, twisting or loading with solder. The fast, reliable QTs let you try out new ideas or rejuvenate old ones with complete safety to your discrete and IC components and your fingers. **SIMPLE MOUNTING** Mounting holes in the housing let you top mount to any flat surface with 4-40 flat head screws, or 6-32F self-tapping screws for behind-the-panel mounting. **ACCEPTS ALL STANDARD COMPONENTS** QTs conform to 1/10" grid and are DIP compatible. ICs, diodes, resistors, capacitors, transistors, etc. are plugged right into the socket and/or strip, without messy, time-consuming soldering. **HOOK-UP** All you need is a wire stripper and #22 AWG solid hook-up wire. Connect power and ground leads to your bus strip. Plug in your ICs, transistors, resistors, etc. Now interconnect components with #22 wire. Connect a signal source to bus strip or directly to input point of your circuit. Each socket has **5 tie points per terminal**; each bus strip has **2 separate rows of interconnecting terminals.** Turn on the power and signal source. Hook-up a scope probe, counter, etc. Then, if you have to add a wire, need another IC or component — just plug it in. You'll never burn your fingers or get solder shorts. And you don't need any special patch cords, either. **INTERCONNECTIONS** Each terminal consists of 5 connected solderless tie points formed from a prestressed, spring loaded noncorrosive alloy to insure secure mechanical and low resistance electrical connections.



2 Buses of Interconnected Terminals — 5 Tie Points Each

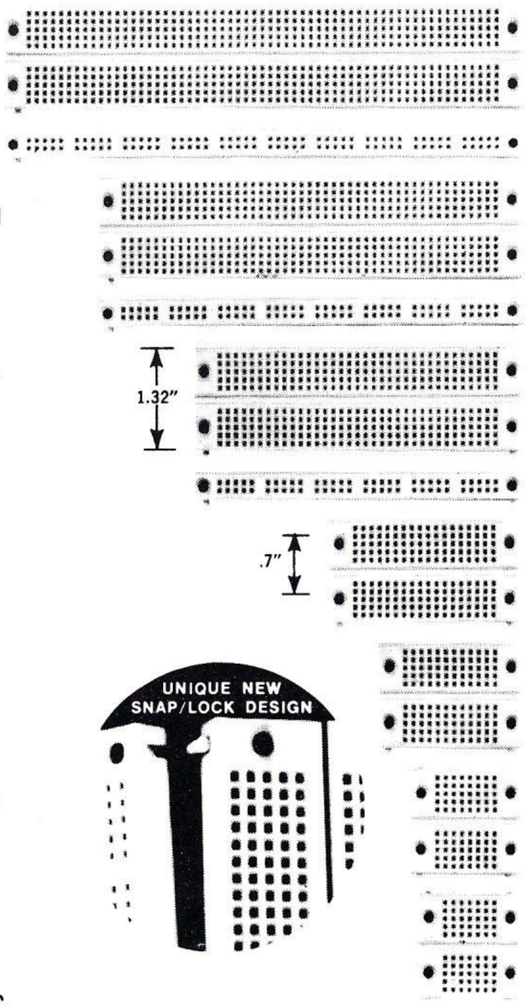
Simply push in your lead, wire or component and you've got a connection, quickly and easily, without damage to either the socket or your components. It's that simple!

All Sockets are 1.32" wide. All Bus Strips are .36" wide. All Sockets and Bus Strips are .33" thick. And all are great for high temperature operation — for worst case circuit design to 100°C.

CONTINENTAL SPECIALTIES

QT SOCKETS & BUS STRIPS*

*QUICK TEST



	Length	Hole-to-Hole	Terminals	Unit Price $
QT-59S	6.5"	6.2"	118	12.50
QT-59B	6.5"	6.2"	20	2.50
QT-47S	5.3"	5.0"	94	10.00
QT-47B	5.3"	5.0"	16	2.25
QT-35S	4.1"	3.8"	70	8.50
QT-35B	4.1"	3.8"	12	2.00
QT-18S	2.4"	2.1"	36	4.75
QT-12S	1.8"	1.5"	24	3.75
QT-8S	1.4"	1.1"	16	3.25
QT-7S	1.3"	1.0"	14	3.00

Made in U.S.A. • Patents Pending
Prices subject to change

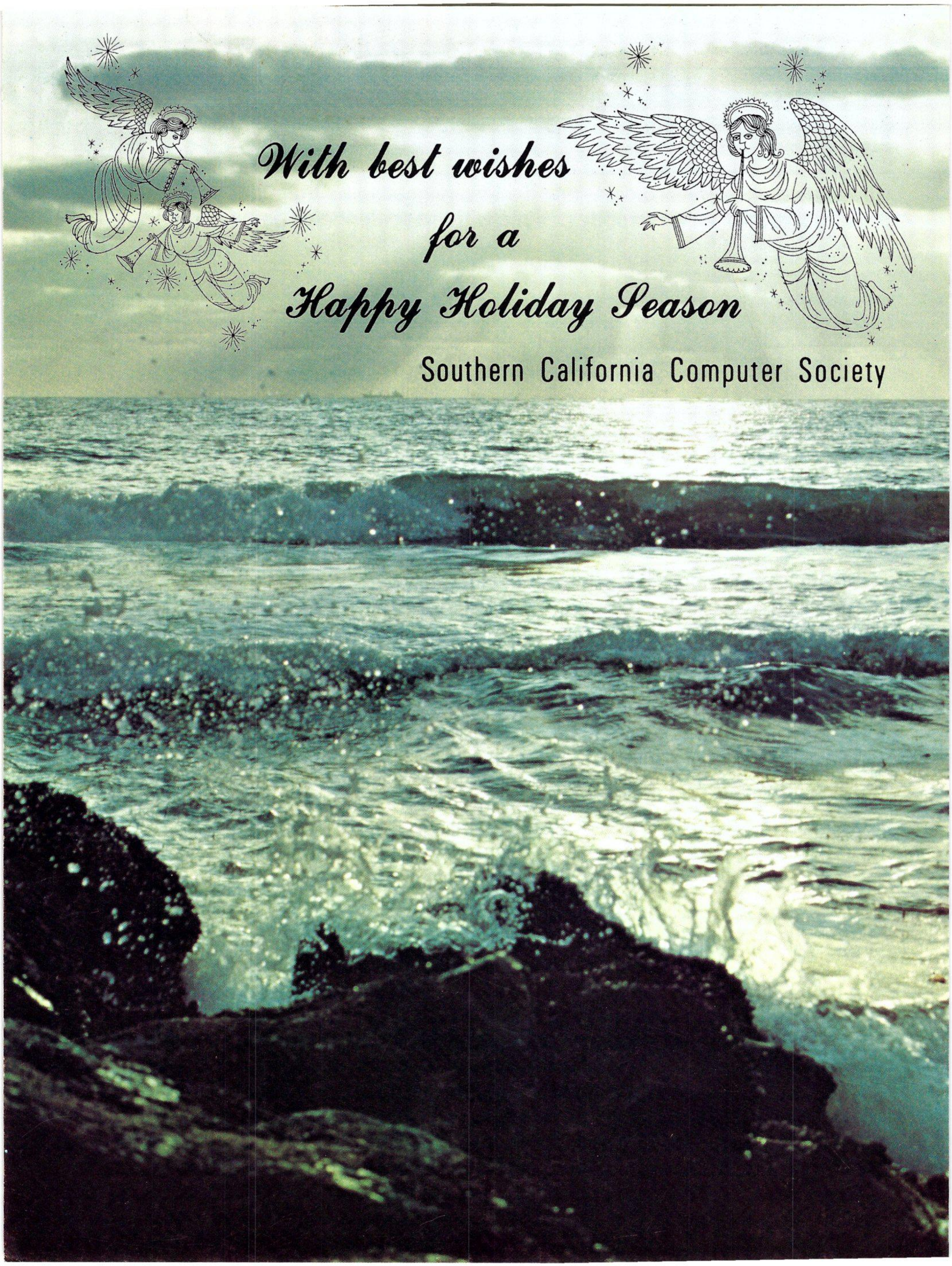
I.C. ELECTRONIC SUPPLY

OPEN 7 DAYS

BankAmericard - UPS - Master Charge

Hours 9 to 9 Mon.-Fri., 9 to 6 Sat. & Sun.

Van Nuys, California 91406 16723 Roscoe Blvd., (213) 894-8171
Orange, California 92667 619 W. Katella Ave., (714) 997-5343
San Diego, California 92111 4861 Convoy St., (714) 292-5484



With best wishes
for a
Happy Holiday Season

Southern California Computer Society